Towards a Knowledge Engineering Methodology for Flexible Robot Manipulation in Everyday Tasks

Michaela Kümpel[1,*], Jan-Philipp Töberg[2], Vanessa Hassouna[1], Philipp Cimiano[2] and Michael Beetz[1]

[1]Institute for Artificial Intelligence, University of Bremen, Bremen, Germany
[2]Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, Bielefeld, Germany

Abstract

In the last decade, there have been great advancements in household robotics, enabling robots to autonomously accomplish household tasks. These robots are typically programmed for specific tasks and/ or objects. We hypothesise that the lack of flexibility in fulfilling new ad-hoc task requests can be overcome by a knowledge-based approach, allowing robots to infer how to address a new task or carry out known tasks on new objects.

Towards this goal, we propose a knowledge-based methodology that leverages knowledge already existing on the web to construct an ontology supporting robots in reasoning about parameters that influence manipulation actions for execution of task variations on a range of objects[1]. The ontology comprises object and action information, covering dispositions and affordances as well as task-specific properties. As a proof-of-concept, we manually construct a food-cutting ontology by importing and linking knowledge from relevant ontologies in addition to extracting and semantically enhancing knowledge from unstructured web sources. We demonstrate how robots can query the ontology and translate the contained information into action parameters. We evaluate the feasibility of the created ontology by simulating a robot accessing the ontology for parameterisation of actions to perform task variations of cutting.

Keywords

Knowledge Engineering, Knowledge Acquisition, Reasoning, Cognitive Robots, Flexible Manipulation

1. Introduction

One of the visions of AI-powered and cognition-enabled robotics are autonomous household robots that can accomplish everyday tasks requiring robots to perform manipulation actions including pouring, cutting or cleaning in natural contexts. Realising robots with such capabilities entails several research challenges for knowledge representation and reasoning. The first challenge is that everyday manipulation tasks, such as “Cut the bread” or “Bring me a coffee”, are typically highly under-determined. The second one is that categories of manipulation tasks are broadly scoped as they include, for example, cutting a large variety of fruits, vegetables, bakery items or meat, with a variety of tools, for a variety of purposes, in many different contexts. Much of the knowledge needed to refine vague tasks and transfer them to new task variations is contained in instruction web sites like WikiHow, encyclopedic web sites like Wikipedia, and many other web-based information sources. The key question is how can we make this abstract knowledge *actionable* for robots?

To answer this question, we examine how modern robot control systems work. Accomplishing a manipulation task can be stated as a reasoning problem: Given a vague task request like “Cut the bread”, infer how the robot has to move its body in order to achieve the goals implied by the request and avoid causing unwanted side effects. In the case of cutting bread, we want the robot to create slices of a desired size and shape, the cuts should be clean, the rest of the bread should not be damaged, and

[1]More information can be found on our website: https://food-ninja.github.io/WebKat-MealRobot/

Workshop on Actionable Knowledge Representation and Reasoning for Robots (AKR³) at Extended Semantic Web Conference (ESWC), May 27, 2024, Heraklion, Crete, Greece

*Corresponding author.
michaela.kuempel@uni-bremen.de (M. Kümpel); jtoeberg@techfak.uni-bielefeld.de (J. Töberg); hassouna@uni-bremen.de (V. Hassouna); cimiano@techfak.uni-bielefeld.de (P. Cimiano); beetz@cs.uni-bremen.de (M. Beetz)
0000-0002-0408-3953 (M. Kümpel); 0000-0003-0434-6781 (J. Töberg); 0000-0003-1335-5698 (V. Hassouna); 0000-0002-4771-441X (P. Cimiano); 0000-0002-7888-7444 (M. Beetz)

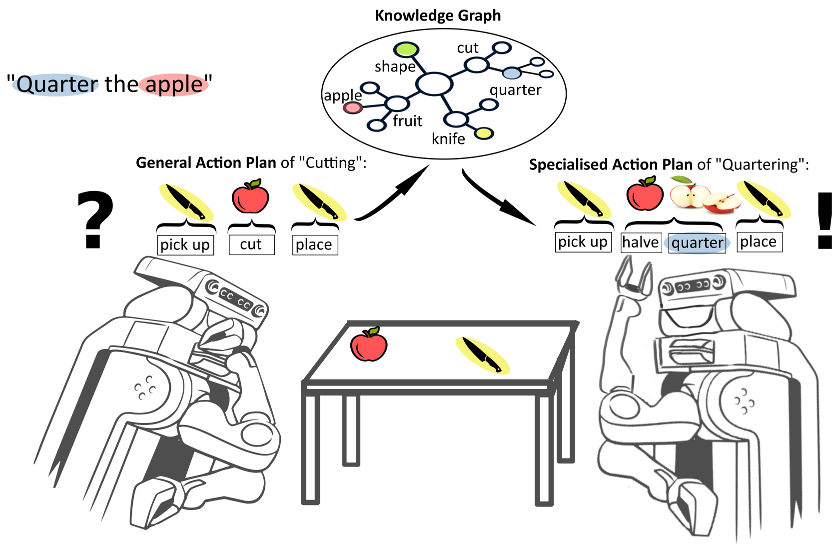


Figure 1: Motivation for this work: Enabling a robot to perform a new task variation of cutting using an ontology.

so on. The body motions of robots can be stated in terms of motion constraints and objectives. For example, when picking up a knife, constraints are to hold the knife by its handle and to keep the blade horizontal, and objectives include generating smooth motions and minimising acceleration to avoid contacting anything other than the target and its underlying surface. The ultimate reasoning task is to infer the constraints and objectives the body motion of the robot should satisfy and maximise given the vague task request. Thus, a result of the reasoning could be to infer the correct motion sequence needed for successful action execution, but it might also include inferring the action parameters needed to successfully parameterise a general action.

We propose to bridge the gap between the information accessible in the web and the knowledge needed by robots to generate perception-guided behaviour by constructing robot-understandable and -processable ontologies of task knowledge, as motivated in Figure 1. To this end, we present a new methodology for knowledge engineering that is capable of creating ontologies formalising action and object knowledge through the concepts of affordance and disposition. We identify manipulation relevant properties of actions and objects that characterise them in terms that make them executable and thus actionable for robots. In a simulation experiment, we demonstrate how a robot using the same cognitive architecture as a real robot can adapt the execution of actions based on knowledge in the created ontology by asking it to perform the new task of "Quartering an apple" and translating the returned query results into parameters that influence robot behaviour while performing the cutting motion. In this way, we show how actions performed by the robot can be directly parameterised by knowledge in the ontology.

The concrete contributions to our state of understanding in web-enabled knowledge representation and reasoning for robots are the following ones:

- We present a general methodology for knowledge engineering supporting robot manipulation that encodes action and object knowledge in an ontology.
- We show how we have instantiated the methodology in order to model knowledge that is relevant for the manipulation of fruits and vegetables as a proof-of-concept for our approach.
- We show how a robot queries the knowledge base and translates the returned result into action parameters for known motions in a virtual environment.

2. Related Work

Knowledge Engineering for Robotics: The field of knowledge engineering provides multiple, well researched methodologies for the creation of general or domain-specific ontologies and knowledge graphs [1]. However, knowledge engineering methodologies specifically developed for cognitive robots are scarce. The approach by Bermejo–Alonso et al. [2] describes a modular ontology and guidelines for using it as a conceptual framework for future ontology engineering but is focused on the domain of autonomous systems and thus is not suitable to handle the intricacies of robotics. A similar problem occurs with the approach by Prestes et al. [3] that describes a methodology employed by an IEEE working group to create an upper ontology for the robotics domain by collaborating with three subgroups. However, the methodology is still very general and relies on a huge amount of available experts and developers to cover all subgroups. All in all there are no suitable knowledge engineering methodologies focused on creating ontologies for flexible manipulation execution by cognitive robots.

Knowledge Representation for Cognitive Robotics: There are different approaches to representing knowledge for robotics. KnowRob [4, 5] is one of the most influential knowledge and reasoning frameworks in the field of cognitive robotics [6, 7]. Other robotic knowledge frameworks with similar functionalities are ORO [8], OROSU [9], KR^3 [10] and PMK [11]. KnowRob recently got extended by the socio-physical model of activities (SOMA) [12], which defines roles of objects during events, their dispositions as well as affordances for a more flexible task execution [13]. It has previously been shown by Dhanabalachandran et al. [14] how the task of cutting bread can be flexibly adapted if the disposition of an object does not meet the affordance of another object required for task execution.

Executing Task Variations: Robots need a cognitive architecture for successful task planning and execution. Some cognitive architectures for robots are SOAR [15] and iCub [16], but in this work we use the cognitive robot abstract machine (CRAM) [17], which has successfully been used for robots executing household activities such as cooking [18] and setting the table [19].

Correctly executing task variations is still a big challenge in robotics due to the fact that tasks are often underspecified and assume commonsense knowledge about objects and the environment, which is hard to extract and represent [20]. There are various approaches focusing on task sequences and enabling agents to infer the next step, using image recognition [21], natural language processing [22, 23], large language models [24, 25, 26] or plan projection for vaguely defined, known tasks [27] that can be used to infer the next action that needs to be performed only if the action itself is already known to the agent. These approaches are based on the idea that actions are sequences of known motions instead of general plans that can be parameterised.

3. Methodology Overview

We propose a five step methodology for creating ontologies that can be accessed by robots for the autonomous execution of task variations. The complete methodology is visualised in Figure 2 and the in- and outputs for all five steps are summarised in Table 1. All artefacts and supporting materials are available in our GitHub repository[1]. We demonstrate the usability of this methodology for the exemplary task of *Cutting Fruits and Vegetables* in Section 4.

3.1. Step 1) - Define Motion Parameters

The first step necessary for creating a flexible robot manipulation ontology is to define the parameters that influence the execution of an action. These parameters highly depend on the domain of interest. For cutting actions the cutting position or the number of repetitions are important, whereas for a pouring action the viscosity and containment size influence the pouring angle.

[1]GitHub repository: https://food-ninja.github.io/WebKat-MealRobot/

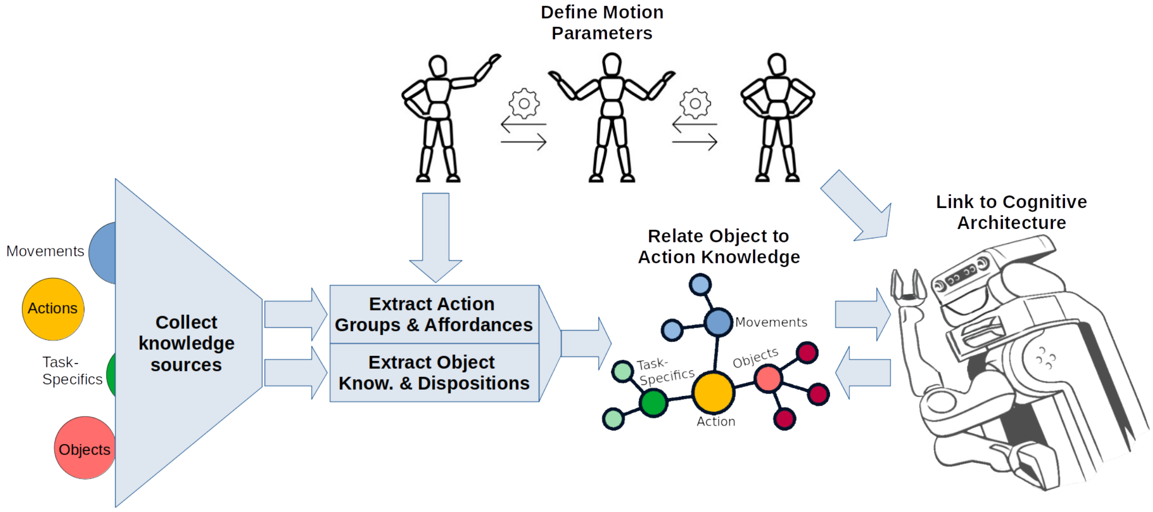


Figure 2: The complete proposed knowledge engineering methodology.

Table 1
Summarising the in- and outputs for each step of the proposed methodology.

Step	Input	Output
1)	Requirements/manipulation plan	List of motion parameters
2)	Requirements & parameters	List of potential sources
3a)	Sources for action knowledge	Action groups, their properties & affordances
3b)	Sources for object knowledge	Objects, their properties & dispositions
4)	Action & Object Knowledge	Ontology
5)	Ontology	Actionable Ontology

The motion parameters have to be aligned with the cognitive architecture of the robot. This work uses CRAM and its general action plans. The idea behind these general action plans is that a wide range of actions can be broken down into the same set of body movements. If we consider the **cutting** action, it can be broken down into the tasks of **picking up**, **cutting** and **placing**, which can further be broken down into the body movements of **approaching**, **grasping**, **lifting** (picking up task), **approaching**, **lowering**, **lifting** (cutting task) and **approaching**, **lowering**, **lifting** (placing task). The same body motion of approaching needs to be parameterised so that for the picking up task, the tool to be used for cutting is approached while for the cutting task the object to be cut shall be approached and for the placing task a placing position is to be approached. While CRAM can resolve such parameters for optimal positions (like the placing position), it is also able to query an ontology when it can not resolve a parameter locally.

3.2. Step 2) - Collect Knowledge Sources for the Given Task

After defining the parameters that influence an action, knowledge sources for the three types of relevant web knowledge are collected. In general, we differentiate between i) action knowledge, ii) object knowledge and iii) knowledge linking the two other knowledge types.

i) *Action Knowledge*: Action knowledge consists of action verbs that are commonly used in a specific domain, as well as properties of a specific manipulation action that are necessary for successfully completing the manipulation task. Additionally, grouping of similar actions into *action groups* makes the contained knowledge executable since it can be reasoned about for parameterisation of actions to achieve task variations.
ii) *Object Knowledge*: As the name suggests, this knowledge contains all relevant information about the objects involved in the task execution (e.g. tools, containers, targets) and their usage. On a

foundational level, this includes any properties or features that are relevant for grounding basic manipulation actions like grasping, holding and transporting to support basic planning. This knowledge supports the robot in understanding and recognising objects and their purpose during task execution. Additionally, *task-specific object knowledge* about object properties relevant for the current manipulation action needs to be collected.

iii) *Knowledge Linking*: To connect and link the action and object knowledge, we rely on the concepts of *disposition* and *affordance*. In general, a *disposition* describes the property of an object, thereby enabling an agent to perform a certain task [28] as in *a knife can be used for cutting*, whereas an *affordance* describes what an object or the environment offers an agent [29] as in *an apple affords to be cut*. In works like [13, 12], both concepts are set in relation by stating that dispositions allow objects to participate in events realising affordances, which are more abstract descriptions of dispositions.

The aforementioned knowledge is collected using multiple types of sources, including (un-)structured sources [30] as well as large language models (LLMs). A collection of exemplary sources focused on commonsense and task-specific knowledge can be found in [20]. Of course, manually creating the ontology with the help of domain experts is possible, but in this paper we are concerned with the extraction of knowledge from existing datasets.

3.3. Step 3a) - Create Action Groups and Extract Affordances

In this step, information about the manipulation action is collected. To cover more use cases and get a better understanding of the action, structured sources are used to collect synonyms and hyponyms of the main verb. By looking at lexical resources like WordNet [31] and VerbNet [32] as well as semantic resources like FrameNet [33], other action verbs with a similar meaning or usage are extracted. To evaluate the relevance of these collected verbs, we propose to analyse unstructured, instruction-focused corpora like WikiHow[2] or, for the cooking domain, Recipe1M+ [34]. WikiHow has the advantage that its articles describe tasks with different levels of granularity, covering high-level (“How to Make an Apple Pie") as well as low-level task instructions necessary to achieve the high-level tasks (“How to Core Apples") [35].

After filtering the initially collected verbs based on their relevance, an analysis of the motion patterns connected to each verb is performed. This analysis focuses on the differences between action execution and how these differences can be constituted as different values for the motion parameters. As an example, consider the verbs “Halving" and “Slicing" which, among other things, differ in the initial position where a cut is applied. For “Slicing", the cutting tool is placed at the end of the target object whereas the tool is placed in the middle for “Halving". Based on these differences, we propose to create *action groups* that summarise all action verbs with a specific set of motion parameters. Now the generalised action plan is crafted that leaves room for these motion parameters to cover all extracted action groups. After defining the different action groups, an affordance for each representative is created. This affordance specifies the task to perform as well as the tool to use for performing the task.

3.4. Step 3b) - Extract Object Knowledge and Dispositions

In addition to the action knowledge, this step collects all relevant information about the participating objects. This includes knowledge about tools and supporting objects (e.g. cutting board) as well as parts of the environment (e.g. drawers) and depends on the scope of the manipulation task to perform. So information about environment parts are not necessary when the robot is operating at a static position where all necessary objects are in grasping range. We propose to focus on a specific category or group of objects (e.g. “Cutting *fruits & vegetables*", “Pouring *liquids*"). Relevant objects from these groups are not chosen manually but extracted from domain-specific object taxonomies, if available. Additional filtering regarding their relevance is done with instruction-focused corpora like WikiHow or Recipe1M+ [34].

[2]www.wikihow.com/Main-Page

After collecting all relevant objects, the relevant object properties are selected. These properties correlate with the aforementioned task-specific object knowledge since they depend on the manipulation task to be performed. After choosing the properties relevant for executing the intended manipulation tasks, the property values for each object are extracted as well. Since this knowledge is often very specific or rooted in commonsense, this extraction is not straightforward and may rely on specific corpora or LLMs [20]. Lastly, based on the relevant object properties, dispositions are created for each object. These disposition highlight the properties that constitute differences in the task execution and influence the motions necessary for successfully performing the manipulation task. For example, if the robot wants to cut a fruit that has a peel, the task to remove the peel needs to be performed before additional cutting is executed.

3.5. Step 4) - Relate Object to Action Knowledge

In this step, the user links affordances to dispositions. For this connection in the TBox, we propose to use the *affordsTask, affordsTrigger* and *hasDisposition* relations introduced in the SOMA ontology [12], as exemplified in Figure 3.

hasDisposition $some$
    $(Cuttability$
    and $($*affordsTask* $some\ CuttingAction)$
    and $($*affordsTrigger* $only\ (classifies\ only\ Knife)))$

Figure 3: Exemplary class expressions showing the connection between a disposition (here: "Cuttability"), the afforded action (here: "CuttingAction") and the tool used for the afforded action (here: "Knife").

3.6. Step 5) - Link Ontology to Cognitive Architecture

Lastly, the created ontology is connected to the cognitive architecture employed for task and motion planning. The motion parameters, extracted action groups and the generalised manipulation plan(s) are considered and different concepts in the plan are mapped to their representation in the ontology. Similarly, the objects and their properties are grounded through the perception system. This is a crucial and difficult part for many cognitive architectures. For the CRAM cognitive architecture for example, this step includes the inference of action parameters at runtime to specialise and contextualise a generalised action plan, which is called an *action designator*. Each designator is connected to a *designator resolver* that refines these generalised instructions into highly specific directives the robot can follow precisely. When the default resolver faces challenges due to missing data or incomplete knowledge, a custom resolver comes into play. One such custom resolver is a knowledge base interface, enabling the robot to seek the missing knowledge necessary for task execution in e.g. an ontology.

In order for a robot to be able to use an ontology as a designator resolver, the returned query parameters need to be available in the general action plan. If the robot shall additionally be enabled to perform task variations, the action plan needs to be modified to allow for the parameterisation of motions.

4. Food Cutting Ontology

In this section we describe how we manually instantiated the methodology to develop a `food_cutting` ontology as a proof of concept for the feasibility of our methodology. The created ontology comprises information relevant for cutting certain fruits and vegetables as well as bread. The information on how to cut bread is partly extracted from the work by Dhanabalachandran et al. [14].

Define Motion Parameters Research suggests that a small number of 15 action verbs occur in more than 50% of all actions in WikiHow instructions [36]. For each of these action verbs, it is crucial to examine the differences in motion and behaviour. All movements in the general action plan are parameterised locally to search for optimal positions but if the motions are to be parameterised by an ontology, the parameters returned by queries to the ontology need to be available as variables in the action plan. For the cutting task, we identify the *cutting position* and the *number of repetitions* to strongly influence action execution. As mentioned before, a halving position will differ from a slicing position. In the same manner, the number of cuts performed for halving is different to the number of cuts performed in slicing.

Collect Knowledge Sources: For the domain of *"Cutting fruits and vegetables"*, we rely on the SOMA ontology [12] to describe general movement and manipulation concepts. For additional action knowledge, we use unstructured sources like cooking videos (e.g. [37]), recipes (e.g. Recipe1M+ [34]) or how-to instructions (e.g. WikiHow corpus [38]). For the object knowledge about fruits and vegetables, we employ parts of FoodOn [39] and the Plant Ontology [40] as a taxonomy, but we use unstructured sources like biology textbooks [41] and cooking videos [37] to gather details about their task-specific properties.

To link these different sources, we reuse terms of the upper ontology of SOMA, as it is based on Dolce+ D&S Ultralite (DUL) [42]. In DUL, an action executes a task and can have a physical object as a participant. From SOMA we know that cutting is a task. We add the information that fruits and vegetables from the FoodOn taxonomy are physical objects. A cutting action in SOMA can be executed by the robot as it translates the cutting action into subtasks which are broken down into motions. From the ontology, the robot can further infer some of the action parameters needed for performing the task, e.g. what tool to grasp, which object to cut, the cutting position or the number of repetitions needed for the specific cutting action.

Table 2

Analysing the occurrences of 18 different hyponyms for "cut" in the WikiHow articles. The three verbs with the most occurrences in each column (despite "cut") are marked **in bold**. We cover *cut* and all its hyponyms except for *crosscut, fillet, sever, slash* and *slit*. For the covered hyponyms, we also state the resulting action group (AG).

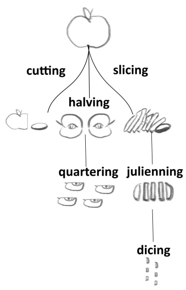
Verb	Title	Method	StepHeader	Sentences	Σ	AG
cut	945	865	2384	8575	12769	cutting
carve	**130**	**103**	53	171	457	cutting
crosscut	0	0	0	2	2	-
chop	21	50	**1002**	**4236**	**5309**	dicing
cube	12	85	23	133	253	dicing
dice	16	**92**	94	611	813	dicing
fillet	10	4	3	17	34	-
halve	16	25	30	119	190	halving
julienne	**38**	23	10	39	110	julienning
mince	12	20	**148**	**937**	**1117**	dicing
pare	2	14	87	412	515	cutting
quarter	0	10	8	44	62	quartering
saw	0	0	11	66	77	cutting
sever	0	0	0	16	16	-
slash	0	0	4	8	12	-
slice	**233**	**371**	**818**	**2637**	**4059**	slicing
slit	0	0	4	8	12	-
sliver	0	0	5	16	21	slicing
snip	0	0	13	74	87	slicing
Σ	1435	1662	4697	18121	25915	
Σ_{AGs}	1425	1658	4686	18070	25839	
Cov. [%]	99.30	99.76	99.77	99.72	99.71	

Creating Cutting Action Groups: We begin by gathering hyponyms for *"Cutting"* from FrameNet [33], WordNet [31] and VerbNet [32] and filter them according to their suitability for the cooking domain. We manually exclude all hyponyms with a meaning that is not applicable (i.e. "amputate") before filtering them based on their occurrences in the WikiHow corpus [38]. For each verb, we look for its present and past tense ("dice" & "diced") as well as its participle form ("dicing"). By using the Part-of-Speech-Tagger of the Stanford Parser [43], we verify that each occurrence is in fact a verb.

Based on these results, we create *action groups* spanning most of the collected verbs. This has the advantage that verbs with few occurrences (e.g. *cube* occurs in only 133 sentences) can be included in our ontology with little additional work. By doing so, we create six action groups covering 13 different verbs. These verbs cover 99.72% of sentences in which a hyponym of "Cutting" occurs and roughly 6% of all instructions in WikiHows "Food & Entertaining" category. The analysis of occurrences of cutting actions in WikiHow and their respective action groups are shown in Table 2.

We differentiate the action groups based on five action properties: the cutting position, the number of repetitions, the kind of input object (whole food or previously cut food part), the resulting objects and the dependencies to prior tasks. A summary of these properties can be seen in Table 3.

Table 3
The six different action groups (AG) and their properties.



AG	Pos.	Rep.	In	Out	Pr. Task
Cutting	`slicing`	$= 1$	Food	FoodPart + Slice	-
Slicing	`slicing`	≥ 1	Food	FoodPart + Slice	-
Julienning	`slicing`	≥ 1	Slice	Slice + Stripe	Slicing
Dicing	`slicing`	≥ 1	Stripe	Cube + Stripe	Julienning
Halving	`halving`	$= 1$	Food	2x Halve	-
Quartering	`halving`	$= 1$	Halve	2x Quarter	Halving

Extract Knowledge about Fruits and Vegetables: When cutting fruits and vegetables, we do not only have to consider different knives that have to be used (as done in [14]), but also information about the cuttable objects' anatomical parts. We gather the information about which parts exist and are relevant for the cutting tasks by combining knowledge from biological books about fruit anatomy (e.g. [41]) with instructional videos from the cooking domain (e.g. [37]).

We hypothesise that no matter which anatomical parts are present in a fruit or vegetable, the important influence factor for action parameterization in cooking activities is the parts' *edibility*. On the one hand, both an apple and an orange do have a peel, but peeling is only mandatory for an orange since its peel is inedible[3], while an apple will usually only be peeled if it is specifically stated in the instruction of a recipe. On the other hand, both an apple and an orange do have some form of a core, but since the core of an apple is inedible (i.e. usually removed before eating), we can infer that it has to be removed before eating.

Including task-specific object properties in the ontology enables robots to infer that specific anatomical parts need to be removed during task execution. In general, we distinguish between edible parts (e.g. apple skin), parts that must be removed before eating due to health or taste reasons (e.g. orange peel) and parts that should be removed but can be eaten if necessary (e.g. apple core).

Relate Object to Action Knowledge in the Ontology: We represent all relevant object and task properties as relations in our `food_cutting` ontology. To provide more details about the developed ontology, we present an overview over the amount of created relations in Table 4. To represent the concepts of affordance and disposition, we employ the relations *affordsTask, affordsTrigger* and *hasDisposition* from the SOMA ontology [12]. The relations *hasEdibility* and *hasPart* directly represent the task-specific object knowledge, which is necessary for action execution. Lastly, we employ the

[3]Although sometimes orange peels are used for cooking, they still need to be removed from the orange beforehand.

Table 4
Relations in the created ontology and their amount.

Relation	#	Relation	#	Relation	#
rdfs:*subClassOf*	182	cut:*affordsPosition*	12	cut:*hasPart*	40
soma:*affordsTask*	78	cut:*hasEdibility*	38	cut:*repetitions*	12
soma:*affordsTrigger*	70	cut:*hasInputObject*	14	cut:*requiresPriorTask*	6
soma:*hasDisposition*	78	cut:*hasResultObject*	14		

affordsPosition, hasInputObject, hasResultObject, repetitions and *requiresPriorTask* relations to represent the properties for each action group, as seen in Table 3.

Link Ontology to CRAM We employ the CRAM cognitive architecture [17]. We developed a custom designator resolver for our generalised action designator that acts as a vital link between the robot and the `food_cutting` ontology. The robotic agent can access the ontology either through a SPARQL API[4] or by running the knowledge processing framework KnowRob that can easily call a module[5] querying the ontology using Prolog.

In Figure 4 an example SPARQL query from the set of queries employed by the simulated robot in Section 6 is shown. The query in Figure 4 is performed once before the execution so that the robot knows whether any prior actions need to be executed. The remaining queries can be found in our GitHub repository.

```
PREFIX owl: <http://www.w3.org/2002/07/owl# >
PREFIX cut: <http://www.ease-crc.org/ont/food_cutting# >
PREFIX soma: <http://www.ease-crc.org/ont/SOMA.owl# >
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema# >
SELECT ?priortask WHERE {
    ${verb} rdfs:subClassOf* ?task.
    ?task owl:onProperty cut:requiresPriorTask .
    ?task owl:someValuesFrom ?priortask.
}
```

Figure 4: Exemplary SPARQL query for querying which prior tasks need to be executed for the given task. *${verb}* is a variable which is substituted by the action's IRI during execution.

5. From Manual to Automated Knowledge Extraction

We envision our methodology to be as automated as possible to better support researchers and practitioners alike in preparing ontologies for their cognitive robots working on everyday manipulation tasks. As a step in this direction, we propose a pipeline for automating the knowledge extraction for the exemplary use case of *"Cutting fruits and vegetables"*, for which we manually created the `food_cutting` ontology in Section 4. In future work we want to evaluate different knowledge sources for this specific domain. An overview over the pipeline for our experiment is visualised in Figure 5.

In general, the pipeline focuses on step 3) (cp. Sections 3.3 and 3.4) of our proposed methodology. As can be seen in Figure 5, we differentiate between the action to perform ("Cutting") and the object group ("Fruits & Vegetables") to focus on. For the action knowledge, we first collect synonyms and hyponyms and filter them based on their occurrence in the WikiHow corpus, as we explained in Section 4. All relevant synonyms and hyponyms are then used as a foundation for creating action groups, a step that is currently performed manually and is hard to automate. After the action group-specific properties are collected, the affordances are created.

[4]SPARQL API: http://grlc.io/api/Food-Ninja/WebKat-MealRobot/SPARQLfiles/
[5]KnowRob module: https://github.com/Food-Ninja/knowrob_cutting

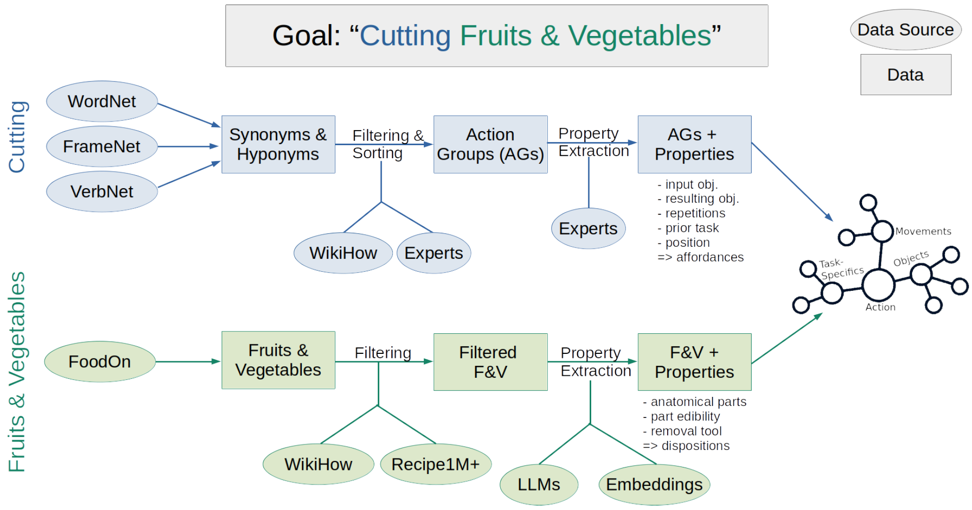


Figure 5: The pipeline for automatically creating the `food_cutting` ontology.

For the object knowledge, we first extract all 257 different fruits and all 31 different vegetables from FoodOn [39]. Of course, not all of these objects are of the same relevance for the cooking domain (e.g. beechnut), so we filter them according to their occurrence in WikiHow articles from the "Food & Entertaining" category as well as the Recipe1M+ corpus [34]. We only include fruits and vegetables that occur in at least 1% of steps in either corpus, which results in 18 remaining fruits and one remaining vegetable. After this filtering step, we try multiple different knowledge sources for gathering knowledge about anatomical parts, their edibility and a fitting removal tool. These sources include word embeddings like ConceptNet Numberbatch [44], GloVe [45] and NASARI [46], large language models like ChatGPT and GPT-4 [47] and filtering based on the Recipe1M+ corpus [34]. Based on the extracted property information, we create suitable dispositions in our ontology. These dispositions can then be linked to the affordances, as explained in Section 3.5.

6. Experiment: Performing Cutting Tasks

To further demonstrate how a robot can use the ontology to parameterise its action plan of cutting, we conducted an experiment where a robot is assigned the task to *"Quarter the apple"*. Simulations offer a method to evaluate robot performance in a controlled yet realistic environment as limitless trials can be run without any risk of damaging the system. The simulated robot is equipped with all software frameworks the real robot would use and also uses CRAM as cognitive architecture. The objects in the simulation are named like ontological concepts (i.e. *knife* and *apple*) and are in direct sight of the robot since we are only interested in successful task execution. In this case, "successful execution" means that the robot achieves the result of a quartered apple or a sliced cucumber without any unexpected or illegitimate motions or states during the execution.

The simulated experiment is shown in Figure 6. Before executing the cutting task, the robot will pose several queries to the `food_cutting` ontology, which we discuss for quartering an apple:

1. What tool should be used for cutting? The returned query result is to use any cutting tool. Since the perceived knife is a cutting tool, the robot will grasp it for cutting the apple.
2. Is any prior action required? The robot will query the ontology for any inedible food parts. The apple has a part that should be removed - its core, which will be returned as query result.
3. Does the given action depend on any prior actions? As described in Table 3, the query will return halving as a prior action for quartering.
4. How many repetitions are needed to perform the action? A halving / quartering action implies that the robot needs to cut just once.

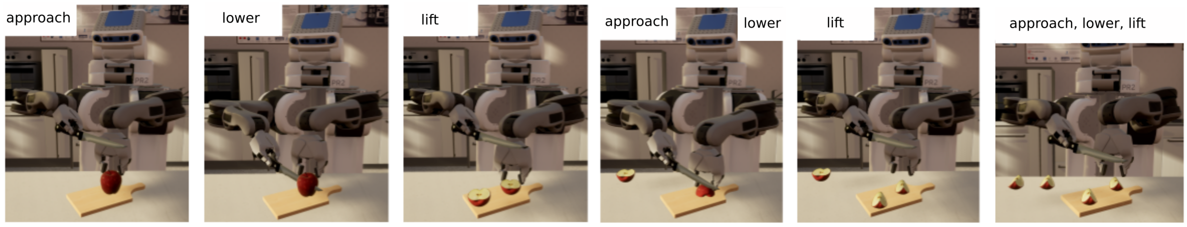


Figure 6: Robot quartering an apple in simulation after querying the ontology for the needed action parameters.

Then, before performing each cutting motion, the robot will query the ontology to retrieve to following information:

1. What is the cutting position that needs to be used? For halving / quartering, the cut should be made in the middle of the object, which is returned from the ontology as *halving_position*.
2. What is the input object to cut? This query is asked at each step (i.e. for halving and quartering, which actually is halving of both halves). For halving, the input object is the food object. When quartering, it is the previously created apple halve that should be halved again.
3. What is the result object of cutting? The robot needs to be aware of what the resulting objects of a cutting action are. Otherwise, it would not be able to differentiate between an apple halve and an apple quarter.

So, to successfully perform the task of *"Quartering the apple"*, the robot would query the manually created ontology eleven times using SPARQL.

7. Conclusion & Future Work

The proposed methodology for knowledge engineering aims to overcome the lack of flexibility in household robotics by leveraging the knowledge that already exists on the web to construct a manipulation-relevant ontology. This ontology supports robots in reasoning on how to manipulate specific objects by providing information about dispositions, affordances and task-specific properties, as well as action groups and their operational properties. As a proof-of-concept, we manually construct a `food_cutting` ontology using web knowledge. We evaluate its feasibility by simulating the task "Quarter an apple" and translating the queried results into parameters influencing cutting motions. In this way, we show how our methodology enables a robot to perform a new task by parameterising a generalised manipulation plan using knowledge from the ontology.

In future work, it will be important to further automate the methodology. We will compare different methods for extracting the relevant knowledge automatically. Some early ideas focus on knowledge extraction from how to corpora and recipes (e.g. WikiHow, Recipe1M+ [34]) as well as using LLMs as a knowledge source (e.g. either directly through prompting or by using OntoGPT [48]). Our focus lies on the creation of the action groups, the extraction of the task-specific knowledge and action properties and the relation of these information in an ontology. Additionally, the methodology should be able to handle food with more complex anatomical parts like stones. Regarding the evaluation of our methodology and the resulting ontologies, we want to investigate further techniques for a quantitative assessment instead of a qualitative simulation, like using question-answering to evaluate the extracted knowledge. Finally, further work is needed on extracting result state information from instructions and mapping them to specific verbs to fully incorporate the verb "Cut" as a general action.

Acknowledgements

The research reported in this paper has been partially supported by the German Federal Ministy of Education and Research; Project-ID 16DHBKI047 "IntEL4CoRo - Integrated Learning Environment for Cognitive Robotics", University of Bremen as well as the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 "EASE - Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject R04 "Cognition-enabled execution of everyday actions". We also want to thank the Joint Research Center on Cooperative and Cognition-enabled AI (https://coai-jrc.de/) for its support.

References

[1] R. Iqbal, M. A. A. Murad, A. Mustapha, N. M. Sharef, An Analysis of Ontology Engineering Methodologies: A Literature Review, Res. J. Appl. Sci. Eng. Technol. 6 (2013) 2993–3000. URL: https://pdfs.semanticscholar.org/c017/bfc3d6c2b7fb3f6d2042b6cd483a63efce87.pdf.

[2] J. Bermejo–Alonso, R. Sanz, M. Rodríguez, C. Hernández, Ontology Engineering for the Autonomous Systems Domain, in: A. Fred, J. L. G. Dietz, K. Liu, J. Filipe (Eds.), Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, volume 348, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 263–277. doi:10.1007/978-3-642-37186-8_17.

[3] E. Prestes, A. Chibani, A. Saffiotti, C. Schlenoff, S. Gérard, R. Sanz, M. E. Barreto, R. Madhavan, Y. Amirat, Towards an upper ontology and methodology for robotics and automation, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, Pittsburgh Pennsylvania, 2012, pp. 873–882. doi:10.1145/2370216.2370414.

[4] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, G. Bartels, KnowRob 2.0 - A 2nd Generation Knowledge Processing Framework for Cognition-enabled Robotic Agents, in: A. Zelinsky, F. Park (Eds.), Proceedings of the 2018 IEEE International Conference on Robotics and Automation, IEEE, Brisbane, Australia, 2018, pp. 512–519. doi:10.1109/ICRA.2018.8460964.

[5] M. Tenorth, M. Beetz, KNOWROB - Knowledge Processing for Autonomous Personal Robots, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, St. Louis, MO, USA, 2009, pp. 4261–4266. doi:10.1109/IROS.2009.5354602.

[6] A. Olivares-Alarcos, D. Beßler, A. Khamis, P. J. Gonçalves, M. K. Habib, J. Bermejo-Alonso, M. Barreto, M. Diab, J. Rosell, J. Quintas, J. Olszewska, H. Nakawala, E. Pignaton, A. Gyrard, S. Borgo, G. Alenyà, M. Beetz, H. Li, A Review and Comparison of Ontology-based Approaches to Robot Autonomy, The Knowledge Engineering Review 34 (2019) 1–38. doi:10.1017/S0269888919000237.

[7] M. Thosar, S. Zug, A. M. Skaria, A. Jain, A Review of Knowledge Bases for Service Robots in Household Environments, in: Proceedings of the 6th International Workshop on Artificial Intelligence and Cognition, CEUR-WS.org, Palermo, Italy, 2018, pp. 98–110. URL: https://ceur-ws.org/Vol-2418/paper11.pdf.

[8] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, M. Beetz, ORO, a knowledge management platform for cognitive architectures in robotics, in: R. C. Luo, H. Asama (Eds.), Proceedings of the 2nd IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Taipei, Taiwan, 2010, pp. 3548–3553. doi:10.1109/IROS.2010.5649547.

[9] P. J. Gonçalves, P. M. Torres, Knowledge representation applied to robotic orthopedic surgery, Robotics and Computer-Integrated Manufacturing 33 (2015) 90–99. doi:10.1016/j.rcim.2014.08.014.

[10] S. Zhang, M. Sridharan, M. Gelfond, J. Wyatt, KR^3: An Architecture for Knowledge Representation and Reasoning in Robotics, in: Proceedings of the 15th International Workshop on Non-Monotonic Reasoning (NMR 2014), arXiv, Vienna, Austria, 2014. doi:10.48550/ARXIV.1405.0999.

[11] M. Diab, A. Akbari, M. Ud Din, J. Rosell, PMK—A Knowledge Processing Framework for Autonomous Robotics Perception and Manipulation, Sensors 19 (2019) 1166. doi:10.3390/s19051166.
[12] D. Beßler, R. Porzel, M. Pomarlan, A. Vyas, S. Höffner, M. Beetz, R. Malaka, J. Bateman, Foundations of the Socio-physical Model of Activities (SOMA) for Autonomous Robotic Agents, in: Formal Ontology in Information Systems, volume 344 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, Amsterdam, 2022, pp. 159–174. URL: https://ebooks.iospress.nl/doi/10.3233/FAIA210379. arXiv:2011.11972.
[13] D. Beßler, R. Porzel, M. Pomarlan, M. Beetz, R. Malaka, J. Bateman, A Formal Model of Affordances for Flexible Robotic Task Execution, in: Frontiers in Artificial Intelligence and Applications - Volume 325: ECAI 2020, volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, Santiago de Compostela, Spain, 2020, pp. 2425–2432. doi:10.3233/FAIA200374.
[14] K. Dhanabalachandran, V. Hassouna, M. M. Hedblom, M. Kümpel, N. Leusmann, M. Beetz, Cutting Events: Towards Autonomous Plan Adaption by Robotic Agents through Image-Schematic Event Segmentation, in: Proceedings of the 11th Knowledge Capture Conference, K-CAP '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 25–32. doi:10.1145/3460210.3493585.
[15] J. E. Laird, The Soar Cognitive Architecture, MIT Press, Cambridge, MA, 2019.
[16] D. Vernon, C. von Hofsten, L. Fadiga, The iCub Cognitive Architecture, volume 11, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 121–153. doi:10.1007/978-3-642-16904-5_7.
[17] M. Beetz, L. Mösenlechner, M. Tenorth, CRAM - A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments, in: R. C. Luo, H. Asama (Eds.), Proceedings of the 2nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), IEEE, Taipei, Taiwan, 2010, pp. 1012–1017. doi:10.1109/IROS.2010.5650146.
[18] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, M. Tenorth, Robotic Roommates Making Pancakes, in: 2011 11th IEEE-RAS International Conference on Humanoid Robots, IEEE, Bled, Slovenia, 2011, pp. 529–536. doi:10.1109/Humanoids.2011.6100855.
[19] G. Kazhoyan, S. Stelter, F. K. Kenfack, S. Koralewski, M. Beetz, The Robot Household Marathon Experiment, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE Computer Society, Xi'an, China, 2021, pp. 9382–9388. doi:10.1109/ICRA48506.2021.9560774.
[20] J.-P. Töberg, A.-C. N. Ngomo, M. Beetz, P. Cimiano, Commonsense knowledge in cognitive robotics: A systematic literature review, Front. Robot. AI 11 (2024). doi:10.3389/frobt.2024.1328934.
[21] I. G. Ramirez-Alpizar, R. Hiraki, K. Harada, Cooking Actions Inference based on Ingredient's Physical Features, in: 2021 IEEE/SICE International Symposium on System Integration (SII), IEEE, Iwaki, Fukushima, Japan, 2021, pp. 195–200. doi:10.1109/IEEECONF49454.2021.9382721.
[22] D. Nyga, S. Roy, R. Paul, D. Park, M. Pomarlan, M. Beetz, N. Roy, Grounding Robot Plans from Natural Language Instructions with Incomplete World Knowledge, in: Proceedings of The 2nd Conference on Robot Learning (CoRL 2018), volume 87, PMLR, Zurich, Switzerland, 2018, pp. 714–723. URL: https://proceedings.mlr.press/v87/nyga18a.html.
[23] I. Sera, N. Yamanobe, I. G. Ramirez-Alpizar, Z. Wang, W. Wan, K. Harada, Assembly Planning by Recognizing a Graphical Instruction Manual, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Prague, Czech Republic, 2021, pp. 3138–3145. doi:10.1109/IROS51168.2021.9636041.
[24] Y. Ding, X. Zhang, S. Amiri, N. Cao, H. Yang, A. Kaminski, C. Esselink, S. Zhang, Integrating Action Knowledge and LLMs for Task Planning and Situation Handling in Open Worlds, Auton Robot 47 (2023) 981–997. doi:10.1007/s10514-023-10133-5.
[25] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, A. Zeng, Code as Policies: Language Model Programs for Embodied Control, in: 40th IEEE International Conference on Robotics and Automation (ICRA), IEEE, London, UK, 2023, pp. 9493–9500. doi:10.1109/ICRA48891.2023.10160591.
[26] J.-P. Töberg, P. Cimiano, Generation of Robot Manipulation Plans Using Generative Large Language

Models, in: 2023 Seventh IEEE International Conference on Robotic Computing (IRC), IEEE, Laguna Hills, CA, USA, 2023, pp. 190–197. doi:10.1109/IRC59093.2023.00039.

[27] G. Kazhoyan, M. Beetz, Executing Underspecified Actions in Real World Based on Online Projection, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE Computer Society, Macau, China, 2019, pp. 5156–5163. doi:10.1109/IROS40897.2019.8967867.

[28] M. T. Turvey, Ecological foundations of cognition: Invariants of perception and action., in: H. L. Pick, P. W. van den Broek, D. C. Knill (Eds.), Cognition: Conceptual and Methodological Issues., American Psychological Association, Washington, 1992, pp. 85–117. doi:10.1037/10564-004.

[29] M. H. Bornstein, J. J. Gibson, The Ecological Approach to Visual Perception, The Journal of Aesthetics and Art Criticism 39 (1980) 203. doi:10.2307/429816. arXiv:10.2307/429816.

[30] P. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web Technologies, Chapman & Hall/CRC Textbooks in Computing, CRC Press, Boca Raton, 2010.

[31] G. A. Miller, WordNet: A Lexical Database for English, Communications of the ACM 38 (1995) 39–41. doi:10.1145/219717.219748.

[32] K. K. Schuler, VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon, Ph.D. thesis, University of Pennsylvania, 2005.

[33] C. F. Baker, C. J. Fillmore, J. B. Lowe, The Berkeley FrameNet Project, in: Proceedings of the 36th Annual Meeting on Association for Computational Linguistics -, volume 1, Association for Computational Linguistics, Montreal, Quebec, Canada, 1998, p. 86. doi:10.3115/980845.980860.

[34] J. Marín, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, A. Torralba, Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2021) 187–203. doi:10.1109/TPAMI.2019.2927476.

[35] S. Zhou, L. Zhang, Y. Yang, Q. Lyu, P. Yin, C. Callison-Burch, G. Neubig, Show Me More Details: Discovering Hierarchies of Procedures from Semi-structured Web Data, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2998–3012. doi:10.18653/v1/2022.acl-long.214. arXiv:2203.07264.

[36] D. Nyga, M. Beetz, Everything Robots Always Wanted to Know about Housework (But were afraid to ask), in: A. T. de Almeida, U. Nunes, E. Guglielmelli (Eds.), Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012), IEEE, Vilamoura, Algarve, Portugal, 2012, pp. 243–250. doi:10.1109/IROS.2012.6385923.

[37] Epicurious, How To Slice Every Fruit | Method Mastery | Epicurious, 2019. URL: https://youtu.be/VjINuQX4hbM.

[38] L. Zhang, Q. Lyu, C. Callison-Burch, Reasoning about Goals, Steps, and Temporal Ordering with WikiHow, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 4630–4639. doi:10.18653/v1/2020.emnlp-main.374.

[39] D. M. Dooley, E. J. Griffiths, G. S. Gosal, P. L. Buttigieg, R. Hoehndorf, M. C. Lange, L. M. Schriml, F. S. L. Brinkman, W. W. L. Hsiao, FoodOn: A harmonized food ontology to increase global food traceability, quality control and data integration, npj Sci Food 2 (2018) 23. doi:10.1038/s41538-018-0032-6.

[40] P. Jaiswal, S. Avraham, K. Ilic, E. A. Kellogg, S. McCouch, A. Pujar, L. Reiser, S. Y. Rhee, M. M. Sachs, M. Schaeffer, L. Stein, P. Stevens, L. Vincent, D. Ware, F. Zapata, Plant Ontology (PO): A Controlled Vocabulary of Plant Structures and Growth Stages, Comparative and Functional Genomics 6 (2005) 388–397. doi:10.1002/cfg.496.

[41] R. Crang, S. Lyons-Sobaski, R. Wise, Fruits, Seeds, and Seedlings, in: Plant Anatomy : A Concept-Based Approach to the Structure of Seed Plants, Springer International Publishing Imprint: Springer, Cham, 2018, pp. 649–678.

[42] V. Presutti, A. Gangemi, Dolce+ D&S Ultralite and its main ontology design patterns, in: P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, V. Presutti (Eds.), Ontology Engineering with Ontology

Design Patterns: Foundations and Applications, number 25 in Studies on the Semantic Web, AKA GmbH Berlin, Berlin, Germany, 2016, pp. 81–103.

[43] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

[44] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, AAAI 31 (2017). doi:10.1609/aaai.v31i1.11164.

[45] J. Pennington, R. Socher, C. Manning, Glove: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. doi:10.3115/v1/D14-1162.

[46] J. Camacho-Collados, M. T. Pilehvar, R. Navigli, NASARI: A Novel Approach to a Semantically-Aware Representation of Items, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, 2015, pp. 567–577. URL: http://aclweb.org/anthology/N/N15/N15-1059.pdf.

[47] OpenAI, GPT-4 Technical Report, Technical Report, OpenAI, 2023. URL: https://cdn.openai.com/papers/gpt-4.pdf.

[48] J. H. Caufield, H. Hegde, V. Emonet, N. L. Harris, M. P. Joachimiak, N. Matentzoglu, H. Kim, S. A. Moxon, J. T. Reese, M. A. Haendel, P. N. Robinson, C. J. Mungall, OntoGPT, Zenodo, 2023. doi:10.5281/ZENODO.10383405.