Everything Robots Need to Know to Prepare Recipes

Michaela Kümpel[1], Manuel Scheibl[2], Jan-Philipp Töberg[3], Vanessa Hassouna[1],
Philipp Cimiano[3], Britta Wrede[2,3] and Michael Beetz[1]

***Abstract*— This paper addresses the challenge of enabling robots to autonomously prepare any meal by bridging natural language recipe instructions and robotic action execution. We propose a novel methodology leveraging Actionable Knowledge Graphs (AKGs) to map recipe instructions into six core categories of robotic manipulation tasks, termed Action Cores (ACs): cutting, pouring, mixing, preparing, pick & place, and cook & cool. Each AC is subdivided into Action Groups (AGs), which represent specific motion parameters required for task execution. Using the Recipe1M+ dataset [1], encompassing over one million recipes, we systematically analysed action verbs and matched them to ACs via direct matching and cosine similarity, achieving a coverage of 75.2%. For the remaining unmatched verbs, we employed a neuro-symbolic approach, matching verbs to existing AGs or generating new action categories through a Large Language Model (LLM). Our findings highlight the versatility of AKGs in adapting general plans to specific robotic tasks, validated through an experimental application in a meal preparation scenario. This work sets a foundation for adaptive robotic systems capable of performing a wide array of complex culinary tasks with minimal human intervention.**

***Index Terms*— Robot Manipulation, Knowledge Graph, Recipe Analysis, Meal Preparation**

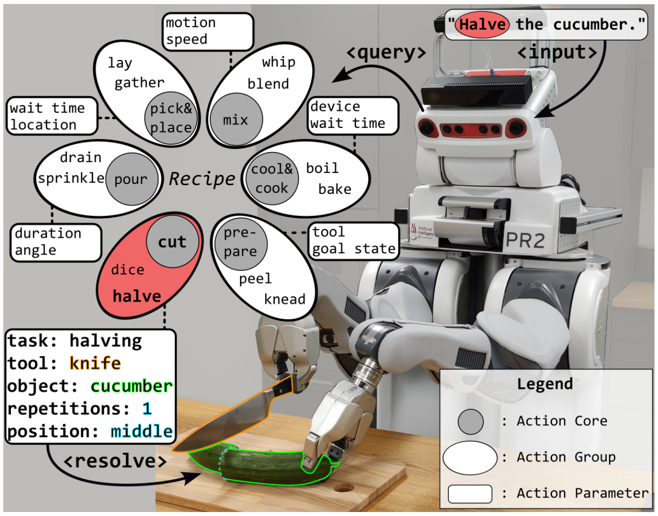

Fig. 1: Motivation of this work: Using Action Cores and Action Groups to parametrise generalised action plans.

I. INTRODUCTION

Robots (still) don't prepare our daily dishes. This is due to the fact that the manipulation skills involved in meal preparation actions are very complex. Even if we consider only one action category like cutting, we have to consider many factors that influence action execution and the desired goal state, such as object properties (e.g. the existence of a peel), task variations (such as halving or slicing) and their influence on motion parameters, as well as the situational context (e.g. the available tools).

In order for robots to be able to successfully compute the body motions needed for execution of different recipe instructions, they need *knowledge*. This work addresses the question how we can build knowledge bases for meal preparation actions that robots can use to translate the contained information into body motion parameters.

While recent research has focused on translating natural language instructions to parameters for pick and place tasks (see [2], [3]), preparation of recipes requires more knowledge than grounded environment information: Recipes contain 1) common sense knowledge such as that a cup sometimes is used as a container but sometimes as a measurement unit, 2) physics knowledge such as that a filled cup should be held upright, as well as 3) manipulation knowledge such as that hot content will burn your finger/ gripper, so a hot cup should be held by its handle.

Thus, accomplishing complex manipulation tasks for the preparation of recipes can be stated as a reasoning problem: Given a list of vague task requests such as *"Set aside for 15 min, then drain and put into a blender"*, infer the **objects** to use based on the text and the current scene graph of the environment, as well as the necessary **body motions** to achieve the desired result while avoiding unwanted side effects. The main question is: **How can we build knowledge bases that represent such knowledge in a machine-understandable way?**

Kümpel [4] proposes a methodology to create *Actionable Knowledge Graphs* as knowledge bases robots can use for action execution. An *Actionable Knowledge Graph* (AKG) connects object information to environment information and action information for an embodiment of knowledge. AKGs provide action parameters for different Action Groups (AGs) of an action category, which can be used in general action plans [5] for the execution of task variations such as for performing slicing, dicing and halving using a general action

[1] Michaela Kümpel, Vanessa Hassouna and Michael Beetz are with the Institute for Artificial Intelligence, Bremen University, Germany {michaela.kuempel, hassouna, beetz}@cs.uni-bremen.de

[2] Manuel Scheibl and Britta Wrede are with the Medical Assistance Systems Group, Medical School OWL, Bielefeld University, Germany manuel.scheibl@uni-bielefeld.de

[3] Jan-Philipp Töberg, Philipp Cimiano and Britta Wrede are with the Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, Germany {jtoeberg, cimiano, bwrede}@techfak.uni-bielefeld.de

plan for cutting [6], [7]. Hence, AGs provide agents with knowledge about how a particular activity shall be performed in a specific context, going along with the awareness about objects involved in these actions as well as the properties that influence task execution. For the example use case of cutting fruits, such an AKG has been used by a robot to infer the necessary body motions for a range of cutting tasks on different objects, from slicing a cucumber to halving an apple [8].

Still, in order for a robot to prepare any recipe given the enormous - and possibly open ended - amount of recipes, the question has to be asked if it is possible to derive such AKGs for each recipe. We break this down to the following questions:

1) How many action verbs, and categories, occur in recipes?
2) Do we need to build AKGs for all of these?

This paper answers these questions by using **Action Cores** (ACs) (an AC is a main manipulation capability like cutting that can be translated to a general action plan), **Action Groups** (AGs) (an AC consists of several more specific AGs that use a similar manipulation plan and thus result in similar body movements and outputs, e.g. the AC of cutting consists of the AGs of dicing, slicing, etc.) as well as an **Actionable Knowledge Graph**, that contains task, object and environment knowledge and enables robots to infer the body motions needed to prepare any given recipe. We hypothesize that recipes consist of six main ACs that can be broken down into several AGs, as visualized in Figure 1. To test this hypothesis, we created AKGs for these six main ACs, analysed the Recipe1M+ corpus [1] consisting of 1,029,720 recipes for the verbs occurring in the recipe instructions and matched them with the verbs of the AKGs.

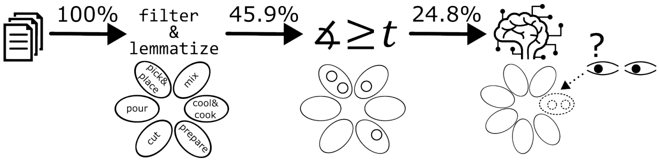

Fig. 2: From the recipe corpus, all included verbs are being matched against the Action Cores and Groups in three steps.

In a first matching step where only direct matches between lemmatized verbs from the recipes and actions in our proposed ACs were considered, we found that the six proposed ACs cover roughly 54% of actions in the corpus. To extend these results, we employed cosine similarity to match all actions above a certain, experimentally defined threshold, bringing the coverage up to $\sim 75.2\%$. For the remaining unmatched verbs, we employed a neuro-symbolic approach, matching verbs to existing AGs or generating new action categories through different Large Language Models (LLMs). The complete pipeline is visualised in Figure 2. The contributions of this paper are the following:

- We define the six main Action Cores for meal preparation tasks.
- We create Actionable Knowledge Graphs for the ACs.
- We perform a neuro-symbolic experiment to match verbs of recipe instructions with our AKGs and categorise verbs that are not covered by our AKGs.

The contributions are validated by letting different simulated robots execute different tasks, in different environments[1].

II. RELATED WORK

Correctly executing unknown tasks 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 [9]. Previous approaches like the work by Forbes and Choi [10] try to infer the implicitly embedded physical knowledge centred around actions and their participating objects. However, for the execution of unknown meal preparation tasks, physical implications focused on size, weight or strength are not conclusive enough to empower robots.

For meal preparation tasks, instructions can be found in recipes, which are structured task sequences written in natural language. To support robotic execution as well as general learning tasks, previous works have focused on analysing different aspects of recipes. For example, the work by Yasukawa and Scholer [11] focuses on the concurrency of dish titles and ingredients whereas the work by Nyga and Beetz [12] analyses actions and their frequency in a WikiHow corpus. Another work by Kiddon et al. [13] focuses on mapping recipe instructions to action graphs describing the participating objects and order of the actions found in the instructions.

Before the execution of meal preparation tasks, robots first need a proper understanding of recipes, which is demonstrated by the benchmark introduced by Nevens et al. [14]. For the correct execution, robots can rely on symbolic planning [15], functional networks combined with task trees [16] or large and vision language models [17]. In our approach, we create action cores and groups and their corresponding knowledge graphs to provide the robot's cognitive architecture with access to situationally relevant knowledge as a basis to parametrise generalised action plans. Our created action representation is hierarchical and similar to the hierarchical action taxonomy by Pereira et al. [18], which focuses on actions performed by service workers in the food industry. However, their work envisions different machines and robots for executing the different actions, whereas we empower a single robot to execute all actions in our cores.

To also include actions not covered by our six action cores, we experiment with a neuro-symbolic approach for classification. Using a LLM to automatically categorise new entities into unknown classes is also proposed by Høeg and Tingelstad [19]. Generally, the topic of automatic sorting bas been explored by e.g. Guerin et al. [20], but in most cases this problem is focused on objects instead of actions.

[1] The experiment is openly available online: https://vib.ai.uni-bremen.de/page/labs/action-cores/

III. FROM RECIPES TO BODY MOTIONS

This work is based on prior work on analysing the amount of different actions occurring in the WikiHow corpus [12], where the authors found that the top 15 action verbs occur in more than 50% of instructions in WikiHow recipes. We go a step further and hypothesize that most action verbs occurring in recipe instructions can actually be broken down into six main ACs that contain AGs. To verify our hypothesis, we analyse a recipe corpus called Recipe1M+ [1]. This dataset contains 1,029,720 recipes with 10,767,598 instructions collected from different sources and written in natural language. Each recipe also contains ingredients and associated food images, but in this work we focus on analysing the verbs that are used in the preparation instructions.

Considering the action verb frequencies, we propose to classify them into the six main ACs of *cutting, pouring, mixing, preparing, pick & place, cook & cool* that comprise of several AGs, which can be translated to motion parameters of robot action plans, as will be explained in the following. These six action cores were proposed in a manual step and afterwards modified, adapted and extended based on the analysis results.

A. Analysing Action Verb Frequencies

The 1,029,720 recipes from Recipe1M+ were used as the input data for the *Spacy* library to assess the dependency trees of the given recipes instruction. For the analysis we used the pre-trained `en_core_web_trf` model[2], which is based on the RoBERTa architecture [21].

In a first step, the recipes and their instructions are parsed by extracting all words that *Spacy* classifies as verbs after part-of-speech tagging, resulting in 22,103,297 words. To further process these verbs, special characters are dropped, a spell check is applied and the verbs are all put into lowercase The remaining verbs are lemmatized with *Spacy* to bring them into their infinitive case. This preprocessing results in 21,876,357 tokens.

At this point the database of verbs does still include duplicates, which are not necessary for the further assessment. Still, a frequency check at this point does provides valuable insights into the most prominent verbs of the recipe set. In Table I the 20 most frequent verbs in the corpus are listed. The fourth column shows the number of times that this particular verb (after pre-processing and lemmatization) is included in the set of recipes. The fifth column lists the relative frequency of a verb in the entire corpus. The 20 most common verbs make up for about 43.62% of all verbs.

B. Action Cores

Looking at Table I, we can identify six main Action Cores that consist of one hypernym for general preparation actions (prepare), three main manipulation actions (cutting, pouring, mixing) and two main categories for tool and device use (pick & place, cool & cook):

TABLE I: 20 most common verbs in the Recipe1M+ corpus [1]. They make up for about 43.617% of all 22,103,297 found verbs.

	Action	**Action Core**	**Frequency**	**Freq. [%]**
1	add	mix	1,573,681	7.120
2	stir	mix	914,835	4.139
3	cook	cook	757,749	3.428
4	place	pick & place	556,027	2.516
5	serve	pick & place	508,347	2.300
6	mix	mix	473,761	2.143
7	bake	cook	460,751	2.085
8	combine	mix	453,977	2.054
9	remove	prepare	450,382	2.038
10	cover	pick & place	443,746	2.008
11	use	pick & place	364,235	1.648
12	heat	cook	356,973	1.615
13	pour	pour	350,273	1.585
14	cool	cool	312,535	1.414
15	remain	-	301,378	1.363
16	set	pick & place	297,351	1.345
17	cut	cut	291,073	1.317
18	sprinkle	pour	260,481	1.178
19	make	-	257,922	1.167
20	preheat	cook	255,503	1.156
$\sum$		18/20 matched	9,640,798	43.617

- **Preparing**: Many recipes include preparation tasks to prepare the food objects for further handling or bringing it into a desired shape. Many of these tasks (e.g. peeling, kneading) are difficult to be performed by a robot, unless they use a tool/ device.
- **Pouring**: Pouring is an action with a simple motion but where the specific task, object properties and ingredient consistency heavily influence motion parameters and successful action execution.
- **Cutting**: Cutting is an action with a complex motion sequence and the goal of dividing an object into two or more pieces of a certain shape. Its execution is influenced by the specific task and object properties.
- **Mixing**: Mixing can result in a range of different motions, some of which require certain tools or containers. Its execution depends on the specific task, available objects, as well as ingredient consistency and temperature.
- **Pick & Place**: Pick & Place tasks have been a research focus. Here, for most tasks the focus lies more on object properties that influence successful grasping or specific locations where the object should be placed.
- **Cook & Cool**: Heating and cooling tasks make up an important AC in meal preparation, but can be broken down into Pick & Place tasks that involve a device (e.g. placing in an oven/ microwave).

The verb list in Table I shows two unmatched verbs: ”remain” and ”make”. We hypothesise that “remain” was falsely classified as a verb although it was used as an adverb in the recipe instructions (as in “add the remaining ingredients”). The word “make” is also not used as an instruction leading to an action but rather as an auxiliary verb (i.e. “To make the filling/ cake, put ...”). Thus, we do not consider these two as action verbs that should be included in the ACs.

[2] `https://spacy.io/models/en`

C. Action Groups

Kümpel [4] as well as Kümpel et al. [6] have proposed to create an Actionable Knowledge Graph for the example use case of cutting fruit. Amongst other things, this AKG acquires knowledge from different sources [7], such as synonyms and hyponyms for "cutting" from WordNet [22], VerbNet [23] and FrameNet [24]. The authors propose to group these verbs into Action Groups of verbs that result in similar motion parameters, and output. As a result, verbs like chopping, mincing and cubing are assigned to the "dicing" AG, which results in different motion parameters than the AGs of Halving, Slicing, or Cutting.

Since the concrete instantiation of parameters is the same for all actions in one AG, using them to cluster similar actions can simplify the actual execution and increase the coverage of novel actions. This makes AGs a crucial influence factor for robots being able to successfully infer and differentiate the body motions for a specific task. Therefore, we reuse the concept of AGs and create them for our six ACs.

D. Action Parameters

As an example, consider this randomly chosen recipe from the dataset for cooking an *Apple And Almond Chutney*:

> **Put** the almonds **into** a small bowl and **add in** sufficient boiling water to **cover** them.
> **Set aside** for 15 min then **drain** and **put into** a blender.
> **Peel** and **core** the apple and **chop** it roughly. **Mix** with the lemon juice and **add in** to the blender together with the remaining ingredients.
> **Blend** till smooth.
> **Refrigerate** for an hour.

With the defined ACs and AGs we can now translate the verbs of the instructions to parameters of the general action plans of the robot, as exemplarily explained by Beetz et al. [8].

The example recipe could thus be translated into the following ACs and corresponding parameters:

TABLE II: Example translation of verbs to action parameters.

Step	Verb	Action Core	Action Parameter
1	**put into**	pick & place	object, destination
2	**add in**	mixing	ingredient, destination
3	**cover**	pick & place	object, destination
4	**set aside**	pick & place	object, destination
5	**drain**	pouring	object, sieve
6	**put into**	pick & place	object, destination
7	**peel**	preparing	object, tool
8	**core**	preparing	object, tool
9	**chop**	cutting	object, tool, position, repetitions
10	**mix**	mixing	ingredients, motion, tool
11	**add in**	mixing	ingredient, destination

E. Towards Preparing any Meal

By adding the action parameters to the AKG, the meal preparation knowledge graph can be used to parametrise generalised manipulation plans, as demonstrated by Kümpel et al. [6]. However, as a next step towards enabling robots to prepare any meal, the abstract parameters incorporated in the knowledge graph need to be grounded in the actual instruction found in the recipe. Previous approaches, like the one proposed by Kanazawa et al. [17], use LLMs to extract the concrete instantiations found in recipe instructions for abstract action parameters. Based on their affirmative results, we also suggest to employ a neuro-symbolic component for the plan parameterization for each concrete recipe to enable robots to perform any necessary meal preparation task.

IV. HANDLING UNMATCHED ACTIONS

With the direct matching of action verbs in the AGs, we were able to match $\sim 54\%$ of the mentioned verb tokens in the corpus. To improve this result, we first find similar verbs by calculating the cosine similarity measure between unmatched verbs and the proposed AGs. With this, we are able to match $\sim 75.2\%$ of verbs in the corpus. The remaining, still unmatched verbs are given to LLMs to create new and potentially missing action cores.

A. Finding Similar Verbs

After analysing the corpus and creating the six ACs with their AGs in Section III, there are still verbs remaining in the dataset that have no connection to our graphs. To handle these verbs, we first use *Spacy*'s `en_core_web_lg` model to calculate the cosine similarity between each unmatched action and all verbs of the six different ACs to find the most similar AG.

To determine the threshold of cosine similarity above which unmatched verbs are matched to their most similar AG, we assessed how many verbs were grouped into some existing AC in absolute numbers in Figure 3. Three important facts can be drawn from the graph:

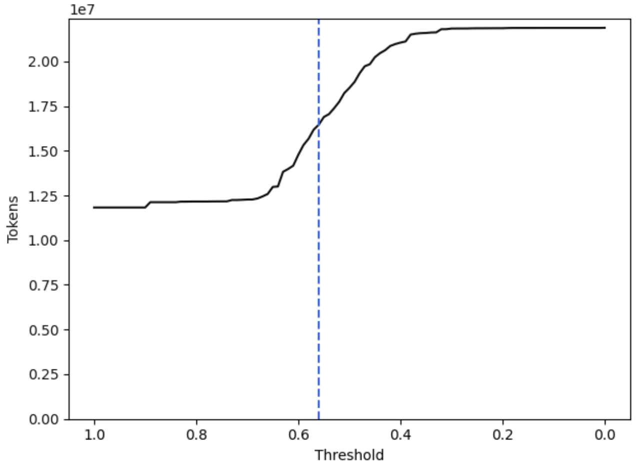

Fig. 3: The line shows the total number of verb tokens from the recipe instructions that would be grouped into ACs based on a threshold for their cosine similarity. The dotted line marks the chosen threshold.

- There is a large number of verbs that fit into the AGs that are included in the knowledge graphs without any cosine similarity applied. 11,826,002 out of all 21,876,357 verbs (∼ 54.06%) were matched directly.
- At a cosine similarity threshold of 0.01, there are still 5,992 verbs ungrouped. This amounts to 334 different verb tokens (5.97% of all found verbs).
- The number of verbs that are grouped to one of the ACs is rising at an applied cosine similarity threshold of 0.56. This indicates that the initially chosen clusters have a high distance to the remaining unmatched verbs.
- Down to a cosine-similarity threshold of 0.64 there is a relatively small rise in grouped verbs. This does indicate that the proposed AGs in the ACs are already covering their respective domain well. If there would be verbs of significant quantity in the corpus that are not included in the ACs but relevant to the domains, there is a high likelihood that they would have been included with a high cosine similarity.

This analysis led us to set the threshold to 0.56. Thereby, 1,206 distinguished verbs were grouped into one of the six ACs. The remaining 4,537 verbs had a smaller similarity, even to their most similar AG.

In Figure 4 the results of matching the verbs from the corpus into the ACs is shown. Moreover, the bars show how many verbs were additionally added to the various ACs by allowing a verb with a minimum cosine similarity of 0.56 to be added to the ACs. What can also be assessed in Figure 4 is the relevance of the different ACs for meal preparation tasks. From the considered ACs, *Cutting* has the lowest presence in the dataset whilst *Mixing* actions have the highest count.

B. Using Large Language Models to Handle Further Verbs

After matching using the cosine similarity, we still remain with 4,537 unmatched verbs that make up roughly 25% of verbs found in the whole recipe corpus. We now want to investigate whether generative LLMs are a suitable source for matching the remaining verbs or coming up with new and currently missing ACs. Since an extensive experiment is beyond the scope of our research, we perform a small pre-study to investigate the general capabilities and assess whether more research in this direction is advisable. To evaluate this pre-study quantitatively, we focus only on a small subset of the remaining unmatched verbs. For this subset we choose the 30 unmatched actions with the most occurrences in the corpus and manually filter them according to two conditions. We exclude:

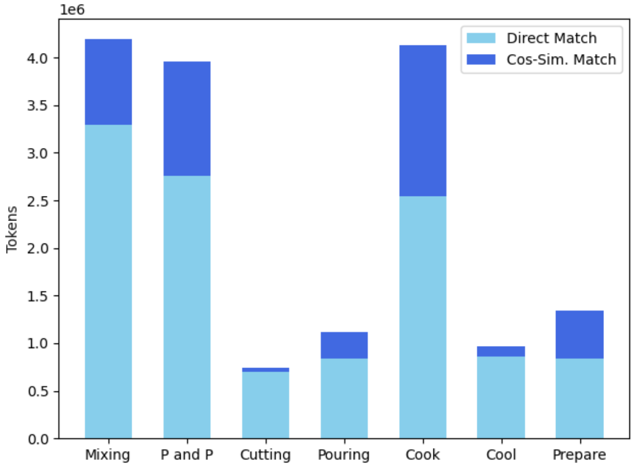

Fig. 4: The number of verb tokens that found a direct match in the ACs and the number of verbs that were added to the ACs by allowing a minimum cosine similarity of 0.56. ”P and P” stands for ”Pick and Place”.

> **System message:** Imagine you are a robot executing meal preparation tasks. I want you to match actions from a given list into pre-defined groups if they entail similar motions during cooking. Each group has one representative action and some associated actions: Action Core → Action Groups
> These are the action cores and groups: *[cores & groups]*
> They differ on the following parameters: *[parameters]*
> **User message:** Match the following actions to one of these action cores through their action groups or create a minimal amount of new cores, if no logical match can be made. If you create a new action core, it should have a similar granularity to the already existing ones. Only match the provided actions and don’t hallucinate new ones.
> Actions to match: *[15 unmatched actions]*

Fig. 5: The prompt given to the LLMs for matching 15 of the remaining unmatched actions to the existing cores or creating new cores, if no suitable match is found.

- Auxiliary verbs (e.g. *make, do, have, let*)
- Abstract verbs that do not describe physical actions (e.g. *remain, need, desire, enjoy*)

After this exclusion, 15 words remain for this pre-study, which we manually mapped to the existing ACs or chose to create a new AC for. These 15 words are the gold standard we compare to.

To perform this pre-study, we query OpenAI’s GPT-3.5 and GPT-4o models and the Claude [25] model five times via their respective API using the prompt in Figure 5. For all five runs of the models, the *temperature* is set to zero to create results that are as deterministic as possible. We calculate the standard classification metrics accuracy, precision, recall, specificity and F1-score and report the results in Table III.

On average, GPT-3.5 slightly outperforms the other models for the accuracy, recall and F1-score, but is outperformed by Claude for precision and specificity. Regarding the proposal of new action cores, the Claude model acts generously, proposing seven new cores on average, whereas GPT-3.5 proposes no new action cores. GPT-4o proposes, on average, one core per run, which is compatible with our gold standard dataset where also only one action is supposed to get a new core (*repeat*). This deliberate proposing of new action cores by Claude is also the explanation for its better performance with regards to precision and specificity. Due to proposing many new cores, Claude creates fewer

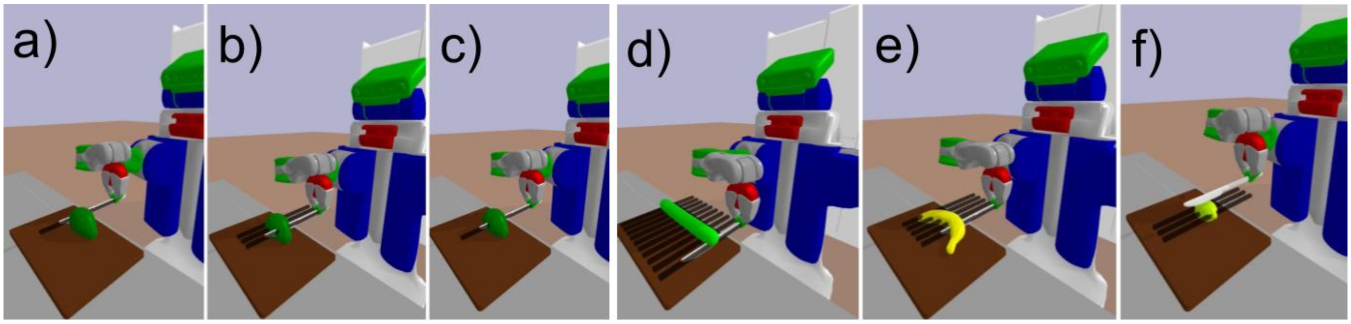

Fig. 6: The PR2 robot demonstrates cutting techniques on various fruits: (a) cutting, (b) slicing, (c) halving an avocado, slicing (d) a cucumber, (e) a banana or (f) a lemon.

mismatches for the remaining verbs, which influences the ratio calculated by the precision metric. Similarly, `Claude` achieves a specificity of 1.0 since it can successfully propose the single expected new action core every run. `GPT-3.5` on the other hand achieves a recall of 1.0 by never proposing any new action cores, thus creating no wrong proposals.

From this small neuro-symbolic pre-study on LLM-based matching, we hypothesise that many of the remaining unmatched actions could, based on their motion-based parametrization, be either mapped into our ACs directly or be decomposed into a combination of the action primitives described by our ACs. Investigating this hypothesis further is part of our future work.

TABLE III: Results for matching the remaining 15 words using three different LLMs and five runs per model. The used models are `gpt-3.5-turbo-0125` (GPT-3.5), `gpt-4o-2024-11-20` (GPT-4o) and `claude-3-5-haiku-20241022` (Claude).

Model	**Run**	**Acc.**	**Prec.**	**Rec.**	**Spec.**	**F1**
GPT-3.5	1	0.60	0.64	1.00	0.00	0.78
GPT-3.5	2	0.67	0.71	1.00	0.00	0.83
GPT-3.5	3	0.67	0.71	1.00	0.00	0.83
GPT-3.5	4	0.67	0.71	1.00	0.00	0.83
GPT-3.5	5	0.67	0.71	1.00	0.00	0.83
GPT-4o	1	0.80	0.86	1.00	0.00	0.92
GPT-4o	2	0.60	0.57	1.00	1.00	0.73
GPT-4o	3	0.53	0.54	0.88	1.00	0.67
GPT-4o	4	0.60	0.62	0.89	1.00	0.73
GPT-4o	5	0.73	0.77	0.91	1.00	0.83
Claude	1	0.53	0.88	0.54	1.00	0.67
Claude	2	0.53	0.88	0.54	1.00	0.67
Claude	3	0.53	0.88	0.54	1.00	0.67
Claude	4	0.53	0.88	0.54	1.00	0.67
Claude	5	0.53	0.88	0.54	1.00	0.67
GPT-3.5	Ø	**0.66**	0.70	**1.00**	0.00	**0.82**
GPT-4o	Ø	0.65	0.67	0.94	0.80	0.78
Claude	Ø	0.53	**0.88**	0.54	**1.00**	0.67

V. PLAN PARAMETRIZATION AS A CORE MECHANISM IN ADAPTIVE ROBOTICS

This section presents a clear example to demonstrate how plan parametrization supports adaptive behaviour in robotic systems. By focusing on a straightforward cutting task, we highlight the process of translating general plans into executable actions. Other ACs can be examined on our website[3]. Plan parametrization plays a key role in enabling robots to adjust their behaviour dynamically. Traditional systems rely on fixed instructions, limiting flexibility. In contrast, our method leverages queries to AKGs to refine task execution in real time. This allows the robot to adapt to varying conditions without requiring constant human intervention, supporting more flexible and efficient automation.

Figure 6 demonstrates the robot's proficiency in adjusting its execution plan for tasks like halving and slicing without prior knowledge of the objects, relying solely on information retrieved from the knowledge graph during execution. The cutting scenario can be tested with all verbs available in the ACs, and on a range of objects, as also detailed in [6]. Our findings underline the robot's adaptability across a range of cutting scenarios, setting a foundation for further advancements. Additionally, our entire setup is available online and accessible to individuals[4].

This example serves as a starting point for examining more complex capabilities. By isolating the key process of plan parametrization, we set the stage for a deeper exploration of the system's computational architecture. This structured approach underscores the importance of plan parametrization in creating adaptable robotics while connecting the concept to broader advancements in system design and decision-making.

VI. CONCLUSION

Towards the goal of empowering robots to successfully prepare any given meal, in this paper we introduce six Action Cores that were identified as central manipulation action categories in the analysed recipe corpus. For each AC we include Action Groups that summarise all actions that result in similar motion parameters and similar manipulation outputs. We match the action verbs found in the Recipe1M+ corpus to our ACs and AGs in two steps: First, we match them directly, covering $\sim 54\%$ of all verbs. Afterwards,

[3] Robot experiment: `https://vib.ai.uni-bremen.de/page/labs/action-cores/`

[4] The hands-on code is available at `https://vib.ai.uni-bremen.de/page/labs/actionable-knowledge-graph-laboratory/`

we match the remaining verbs using cosine similarity and an experimentally defined threshold, leading to a coverage of $\sim 75.2\%$. For the remaining verbs, we query a LLMs to match the verbs or provide us with additional ACs that possibly cover the missing verbs, but a brief analysis shows that the newly proposed cores are already incorporated by our proposed cores, underlining their great coverage for meal preparation tasks.

In the future, we want to take the next step towards robots automatically preparing any meal they encounter by including a neuro-symbolic component that extracts the natural language parameters from the actual recipe text to create the concrete parameterization of each action, as we explained in Section III-E. Additionally, we want to perform more robotic experiments to investigate the resilience of the proposed approach as well as the practicability of the ACs. Lastly, we need to investigate further how the remaining unmatched words can be handled and whether they can be e.g. automatically decomposed into sequences of existing ACs or AGs, as hypothesised in Section IV-B.

References

[1] J. Marín, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 187–203, Jan. 2021.

[2] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. ichter, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. T. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K.-H. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *Proceedings of the 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 2023-12-14/2023-12-18, pp. 287–318.

[3] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Task Planning," in *Conference on Robot Learning (CoRL)*. Atlanta, GA, USA: arXiv, 2023.

[4] M. Kümpel, "Actionable Knowledge Graphs - How Daily Activity Applications can Benefit from Embodied Web Knowledge," Ph.D. dissertation, Bremen University, Bremen, Germany, Mar. 2024. [Online]. Available: https://doi.org/10.26092/elib/2936

[5] V. Hassouna, A. Hawkin, and M. Beetz, "Steps Towards Generalized Manipulation Action Plans - Tackling Mixing Task," in *ESWC 2024 Workshops and Tutorials Joint Proceedings*, Heraklion, Crete, Greece, 2024.

[6] M. Kümpel, J.-P. Töberg, V. Hassouna, P. Cimiano, and M. Beetz, "Towards a Knowledge Engineering Methodology for Flexible Robot Manipulation in Everyday Tasks," in *ESWC 2024 Workshops and Tutorials Joint Proceedings*, Heraklion, Crete, Greece, 2024. [Online]. Available: https://ceur-ws.org/Vol-3749/akr3-04.pdf

[7] M. Beetz, P. Cimiano, M. Kümpel, E. Motta, I. Tiddi, and J.-P. Töberg, "Transforming Web Knowledge into Actionable Knowledge Graphs for Robot Manipulation Tasks," in *ESWC 2024 Workshops and Tutorials Joint Proceedings*. Heraklion, Crete, Greece: CEUR-WS, 2024. [Online]. Available: https://ceur-ws.org/Vol-3749/akr3-tutorial.pdf

[8] ——, "Translating Actionable Knowledge Graphs into Robot Action Execution," in *Proceedings of the Workshops at the Third International Conference on Hybrid Human-Artificial Intelligence Co-Located with (HHAI 2024)*. Malmö, Sweden: CEUR-WS, 2024.

[9] J.-P. Töberg, A.-C. N. Ngomo, M. Beetz, and P. Cimiano, "Commonsense knowledge in cognitive robotics: A systematic literature review," *Front. Robot. AI*, vol. 11, 2024.

[10] M. Forbes and Y. Choi, "Verb Physics: Relative Physical Knowledge of Actions and Objects," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada: arXiv, 2017, pp. 266–276.

[11] M. Yasukawa and F. Scholer, "Concurrence of Word Concepts in Cooking Recipe Search," in *Proceedings of the 9th Workshop on Multimedia for Cooking and Eating Activities in Conjunction with The 2017 International Joint Conference on Artificial Intelligence*. Melbourne Australia: ACM, Aug. 2017, pp. 25–30.

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

[13] C. Kiddon, G. T. Ponnuraj, L. Zettlemoyer, and Y. Choi, "Mise en Place: Unsupervised Interpretation of Instructional Recipes," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 982–992.

[14] J. Nevens, R. De Haes, R. Ringe, M. Pomarlan, R. Porzel, K. Beuls, and P. Van Eecke, "A Benchmark for Recipe Understanding in Artificial Agents," in *2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC-COLING 2024 - Main Conference Proceedings*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds., May 2024, pp. 22–42. [Online]. Available: http://www.scopus.com/inward/record.url?scp=85195902064&partnerID=8YFLogxK

[15] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Interpreting and Executing Recipes with a Cooking Robot," in *Experimental Robotics*, J. P. Desai, G. Dudek, O. Khatib, and V. Kumar, Eds. Heidelberg: Springer International Publishing, 2013, vol. 88, pp. 481–495.

[16] M. S. Sakib, D. Paulius, and Y. Sun, "Approximate Task Tree Retrieval in a Knowledge Network for Robotic Cooking," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11 492–11 499, Oct. 2022.

[17] N. Kanazawa, K. Kawaharazuka, Y. Obinata, K. Okada, and M. Inaba, "Real-world cooking robot system from recipes based on food state recognition using foundation models and PDDL," *Advanced Robotics*, pp. 1–17, Oct. 2024.

[18] D. Pereira, A. Bozzato, P. Dario, and G. Ciuti, "Towards Foodservice Robotics: A Taxonomy of Actions of Foodservice Workers and a Critical Review of Supportive Technology," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 3, pp. 1820–1858, July 2022.

[19] S. H. Høeg and L. Tingelstad, "More Than Eleven Thousand Words: Towards Using Language Models for Robotic Sorting of Unseen Objects into Arbitrary Categories," in *Workshop on Language and Robot Learning (LangRob) at the Conference on Robot Learning (CoRL)*, Auckland, New Zealand, 2022. [Online]. Available: https://openreview.net/forum?id=ngaOl34ycz

[20] J. Guérin, S. Thiery, E. Nyiri, and O. Gibaru, "Unsupervised robotic sorting: Towards autonomous decision making robots," *International Journal of Artificial Intelligence and Applications (IJAIA)*, vol. 9, no. 2, 2018.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019.

[22] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[23] K. K. Schuler, "VerbNet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, University of Pennsylvania, 2005.

[24] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet Project," in *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics -*, vol. 1. Montreal, Quebec, Canada: Association for Computational Linguistics, 1998, p. 86.

[25] Anthropic, "The Claude 3 Model Family: Opus, Sonnet, Haiku, Tech. Rep. Claude-3 Model Card, 2024. [Online]. Available: https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf