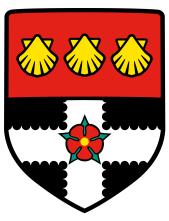
Natural Task Learning through Simultaneous Language Grounding and Action Learning

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Oliver Roesler



Brain Embodiment Laboratory
School of Biological Sciences
University of Reading
UK
November 2021

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Oliver Roesler

Acknowledgments

Our state of mind is a result of not just the temporary input from the environment obtained through our sensors but also the product of the experiences collected since we developed the capability to remember experiences. Thus, this thesis has been formed not just through the last years I have actively worked on it but also during the many years prior to that. Since a lot of environmental input is generated by other agents in the environment, I would like to express my gratitude to all the people I interacted with. Not just the interactions that I consciously remember but also all the other small interactions that had an influence on my mind and therefore also on this thesis. Thus, if we ever interacted with each other, I would like to say thank you for helping shaping this work.

Abstract

Artificial agents and in particular robots, i.e. agents with some form of embodiment, provide nearly unlimited possibilities to support humans in their daily lives by reliably performing hazardous, repetitive, and physically demanding tasks, removing the risk of human errors, and providing social, mental, and physical care as needed, and around-the-clock. However, for this, artificial agents need to be able to communicate with other agents, in particular humans, in a natural and efficient manner, and to autonomously learn new tasks. The most natural way for humans to tell another agent to perform a task or to explain how to perform a task is through natural language. Therefore, artificial agents need to be able to understand natural language, i.e. extract the meanings of words and phrases, which requires words and phrases to be linked to their corresponding percepts through grounding.

Theoretically, groundings, i.e. connections between words and percepts, can be manually specified, however, in practice this is not possible due to the complexity and dynamicity of human-centered environments, like private homes or supermarkets, and the ambiguity inherent to natural language, e.g. synonymy and homonymy. Therefore, agents need to be able to autonomously obtain new groundings and continuously update existing groundings to account for changes in the environment and incorporate new information obtained through the agent's sensors. Furthermore, the obtained groundings should be utilizable to learn new tasks from natural language instructions.

Therefore, this thesis proposes a novel framework for simultaneous language grounding and action learning that achieves three main objectives. First, it enables agents to continuously ground synonymous words and phrases without requiring external support by another agent. Second, it enables agents to utilize external support, if available, without depending on it. Finally, it enables agents to utilize previously learned groundings to learn new tasks from language instructions.

Contents

List of Figures

List of Tables

1 Introduction

Robots are versatile machines that have the potential to revolutionize not only our workplaces, but also our homes. Industrial robots employed in today's factories already outperform humans in terms of physical strength, speed, and precision. However, these robots are deployed in carefully controlled environments and have no or only very limited interaction with humans [42]. In contrast, service robots that are designed to work in complex human-centered environments to support and interact with a variety of other agents, such as humans, pets or other robots, are currently only used to perform simple narrowly defined tasks, such as vacuuming the floor or cutting grass. The main reason is that human environments are intrinsically more complex due to the following characteristics:

- **Other autonomous agents**: Humans, animals, or other robots can be in the same environment so that artificial agents need to be able to interact with them in a natural and efficient manner, which requires artificial agents to be able to converse in natural language. Additionally, they need to adjust their actions based on the conversations, e.g. performing a requested task or adjusting their behavior based on the preferences expressed by the other agents. The latter requires artificial agents to understand the goals and needs of the other agents.
- **Built for humans**: Human-centered environments and the objects within are made for the characteristics and capabilities of the human body, thus, when observing humans perform a task, artificial agents can in most cases not just copy their actions but need to map them to the capabilities of their current embodiment[1]. Furthermore, all actions performed by humans are optimized for the human body, thus, agents need to be able to distinguish relevant constraints that need to be followed to perform the task correctly from constraints introduced by the limitations of the human body to optimize all actions for their own embodiment. Finally, when modifying the environment, artificial agents should ensure that the environment remains suitable for humans, i.e. modifications of the environment should either directly benefit the humans or at least have no negative effect for them.

[1] In contrast to humans, artificial agents can be placed in different embodiments or their embodiments can be modified to extend their capabilities, e.g. by adding additional sensors or actuators to increase the amount of obtainable perceptual information and the number of degrees of freedom, or improve their performance, e.g. by replacing their sensors or actuators with better ones.

- **Assuming common sense**: Human-centered environments and the objects within are designed assuming that the inhabitants have common sense, understand the purpose and characteristics of the objects as well as the way they should be used. Artificial agents need to have similar understanding capabilities to ensure that they do not use objects inappropriately, which could otherwise cause damage to the objects, the agents, or in the worst case other agents in the environment. For example, an agent might throw a knife towards a human, place a baby into a washing machine or put a pet into the microwave, when asked to give a knife to a person, wash a child or dry a pet, respectively. The list of possible dangerous, harmful or even deadly actions is in the end only limited by our imagination but it provides a clear sense of the importance of incorporating common sense and proper *understanding* of the effects of their actions into artificial agents.

- **Dynamicity**: The environment can change due to the actions of other autonomous agents or the dynamicity of nature, such as weather or decay of materials. Understanding these changes and acting appropriately requires artificial agents to be able to determine what or who caused the change, whether it is desired, and whether any supportive or preventive actions should be performed in response. Furthermore, agents should also be aware of the reactions their actions will trigger before performing them, i.e. artificial agents should have models of the other agents' minds[2].

These characteristics make it necessary for artificial agents to build and continuously update their own models of the world so that they are able to understand the environment they operate in, especially the characteristics, needs, actions and goals of other agents, such as humans, to reason about them to plan their actions and achieve their goals. The world models can neither be build purely from abstract or concrete knowledge but need to incorporate both so that the latter grounds the former in the physical world. Abstract knowledge, such as common sense knowledge or knowledge found in books or on the web, is important to enable agents to reason about the world, plan how to achieve their goals and interact appropriately with the environment and other agents. Interacting appropriately does not only mean to prevent physical damage or harm but also includes following of social norms, such as empathy, and taking into account the limitations of other agents to ensure that interactions are experienced as efficient, natural and beneficial by the other agents. Concrete knowledge, on the other hand, is essential to provide meaning to abstract knowledge so that artificial agents can *understand* the current state of the world as perceived through their sensors, to (1) reason about it and determine the actions that have the highest probability of bringing the

[2]Understanding the mental states of other humans is called *Theory of Mind* [67] and is an active research area in psychology.

agents closer to the state of the world they desire, (2) convert the selected actions into actual actuator commands to execute them, and (3) to verify whether the executed actions had the desired effect and whether the agents have achieved their tasks or goals, i.e. whether the current state of the world is the same as the desired state.
Connecting abstract and concrete knowledge is non-trivial and requires sophisticated grounding mechanisms to link abstract concepts, which can have one or more symbolic representations, to their CRs obtained or interpretable by the agent's embodiment. CRs represent sets of invariant features that are sufficient to distinguish perceptual and actuator information belonging to different concepts [37] and can be obtained through any clustering or classification algorithm. Similarly, perceptual and actuator information can take a variety of forms like color histograms, audio signals, or force sensor readings and motor velocity, joint positions, or desired torque, respectively. The main challenge is to obtain a CR that is general enough to cover all possible instantiations of a concept, while at the same time specific enough to avoid confusion with instantiations of other similar concepts. The meaning of the term *concept* is still an area of active philosophical debate (see e.g. [50, 110]) and in most grounding studies it is either used synonymous to words or symbols, e.g. [98, 4], or it is completely avoided by directly stating that words are grounded through CRs, e.g. [57, 51]. Since all the scenarios used in this thesis contain synonyms or homonyms, concepts can neither be directly represented through words nor CRs. Instead the proposed grounding framework represents them implicitly through the connections between words and CRs, which will simplify the integration of an explicit concept representation, in the near future, to enable the use of sophisticated reasoning mechanisms.

This thesis introduces a novel task learning framework which (1) uses both CSL and IL to ground language through corresponding CRs, (2) utilizes the obtained groundings to extract goal states from natural language instructions, and (3) employs RL to learn the actions required to achieve the desired tasks. Both employed grounding mechanisms are inspired by the way children learn the meaning of words. CSL, which belongs to the group of slow-mapping mechanisms through which children learn most words [14], assumes that words and CRs that refer to the same concept co-occur reliably across situations [12, 90] so that they can be identified in an unsupervised manner by looking at co-occurrence information. In contrast, IL, which is a fast-mapping mechanism through which children only learn a small number of words [15, 108], follows the idea that explicit teaching of mappings as well as immediate feedback to confirm correct and remove false mappings increases the speed at which groundings are acquired and reduces the risk of incorrect groundings. The main disadvantage of IL based grounding approaches is that a tutor is required, who might not always be available. Since it is important that agents do not depend on the support of other agents, while it is at the

same time beneficial, if they are able to utilize support when available, the grounding framework presented in this thesis combines both mechanisms.
Additionally, the proposed framework does not require an offline training phase but continuously updates its groundings with every encountered situation to improve the accuracy of previously obtained groundings and to incorporate words and CRs that have not been encountered before. This is important because it is impossible to simulate all theoretically possible situations and train an agent on them before deploying it in the real world[3]. In contrast to most existing grounding frameworks, the proposed framework is also able to handle both synonyms, i.e. multiple words refer to the same concept, and homonyms, i.e. a concept has multiple CRs, which is important because language is inherently ambiguous and many words refer to different concepts depending on the context, e.g. the word *tower* can refer to a high building or a computer case and the word *bridge* can among other things refer to a structure providing a path over an obstacle or the forward part of a ship from which it is navigated.
Finally, the framework has an integrated task learning mechanism that uses obtained groundings to extract goal states from natural language descriptions which are then provided as input to a RL algorithm to learn the correct sequence of actions to perform the desired task. This integration is very important because the main purpose of service robots is to perform tasks requested by humans either alone or in collaboration with other agents and often only a verbal description of the task is available.

1.1 Research Challenges

In this dissertation, three main research challenges are investigated that are essential to enable artificial agents to autonomously connect abstract and concrete knowledge, extract goal states from natural language descriptions, and learn how to perform a desired task with their current embodiments. Each of the research challenges leads to one of the main contributions described in the next section (Section 1.2) and a number of research questions which are introduced at the beginning of each of the chapters addressing one of the challenges and main contributions, i.e. Chapters (3, 4, and 5).

1. **An agent should be capable of grounding ambiguous words in an unsupervised and open-ended manner.** Since the availability of another agent which is able and willing to support the grounding process cannot be guaranteed and to allow the agent to also learn from watching interactions between other agents, the learning agent should be able to learn groundings in an unsupervised manner. Furthermore, the learning agent should be able to learn new words and CRs at any time,

[3] In fact, even if it would be possible to ground all currently existing concepts, it would not be sufficient because new words as well as new concepts that require grounding appear constantly [52, 21].

i.e. no offline training phase should be required, to be able to cope with the dynamic nature of human-centered environments. Finally, the learning agent should be able to handle language ambiguity, i.e. concepts that can be referred to by multiple words that are synonymous in specific contexts, like *big* and *large*, as well as homonymous words that can refer to multiple concepts, like *apple* which can refer to a fruit as well as a company.

2. **An agent should be able to utilize support provided by other agents to improve the acquisition speed and accuracy of obtained groundings.** When another agent is trying to support the grounding process through verbal or non-verbal feedback, the learning agent should be able to utilize the provided support to identify correct groundings faster and discard previously obtained incorrect groundings. However, the agent should not depend on the support because neither its availability nor correctness can be guaranteed. The latter means that it is possible that the information provided by another agent is incorrect, which can be due to malicious intent or by accident, or that the information becomes corrupted during the transfer to the learning agent, e.g. when the learning agent misunderstands to which object the other agent is pointing. Therefore, the provided support should only be used in combination with information obtained in an unsupervised manner to verify the former.

3. **An agent should be capable of utilizing obtained groundings to learn new tasks from natural language instructions.** Enabling an agent to learn new tasks is non-trivial, therefore, most agents are still programmed manually, while the agents that are capable of learning new tasks require enormous amounts of data or very close supervision like haptic demonstrations. Yet, even with this strong support their adaptability is very limited, while obtaining large amounts of data is not always possible or might be more expensive than programming the agent manually for many scenarios and in the end, there is still a large number of cases the agent is not able to handle, thereby, preventing end-user adoption. However, if an agent would be able to learn a new task after getting only a natural language instruction, it would drastically simplify the learning process and enable untrained users to teach the agent new tasks. Furthermore, after the agent successfully learned the task, the natural language instructions can be utilized to explain its behavior to other agents, which is very important to foster acceptance of the agent by untrained users and policy makers, when considering deployment in public places or official functions.

1.2 Contributions

This dissertation contains three main contributions, each addressing one of the research challenges introduced in the previous section (Section 1.1).

1. **A novel framework to ground ambiguous words in an unsupervised and open-ended manner.** In Chapter (3) this thesis introduces a novel framework to ground words in an unsupervised and open-ended manner through their corresponding CRs, which represent sets of invariant features sufficient to distinguish percepts belonging to different concepts. The employed CSL algorithm allows a single word to refer to multiple CRs to handle homonymy as well as multiple words to refer to the same CR to handle synonymy[4]. The framework is evaluated for four different scenarios, which differ based on the used sentences and percepts, to illustrate its applicability to a variety of situations and environments.

2. **A novel framework to combine unsupervised and supervised grounding.** In Chapter (4) the unsupervised grounding framework proposed in Chapter (3) is extended to obtain new and revise existing groundings through feedback to speed up the acquisition of new groundings and improve the accuracy of the obtained groundings by correcting false ones. The CSL component ensures that the framework still works in the absence of any support by a tutor, which is important because the availability of a supporting agent cannot be taken as granted. Furthermore, the supervised learning mechanisms have been integrated into the unsupervised framework so that incorrect supervision has no major negative impact on the acquisition speed and grounding accuracy because the correct groundings will in that case be obtained via CSL. The extended framework is evaluated through two different scenarios, which differ based on the used sentences and percepts.

3. **A novel framework that enables simultaneous, efficient, and unsupervised task learning and grounding.** In Chapter (5) the hybrid framework proposed in Chapter (4) is extended with a mechanism to utilize obtained groundings to extract the goal of a task from a natural language description so that the task can then be learned in an unsupervised manner through RL. If the extracted goal is incorrect, the agent will perform a wrong task, however, if a tutor is available the learned task can be assigned retrospectively to the concept it represents. To evaluate the extension, a task learning simulation is introduced and combined with the two

[4]The framework does not represent concepts explicitly, instead they are implicitly represented through the connections between words and CRs. Most existing grounding frameworks completely avoid language ambiguity, i.e. synonyms and homonyms, and set words equal to concepts, while the few frameworks, e.g. [77, 78] that consider language ambiguity use implicit concept representations similar to the framework presented in this thesis.

grounding scenarios used to evaluate the hybrid grounding framework in Chapter (4).

The thesis is organized as follows. Chapter (2) introduces the necessary background to understand the main contributions of this thesis and discusses existing related research. In Chapter (3) a novel framework for unsupervised and open-ended grounding is proposed and evaluated through four different human-agent interaction scenarios. Afterwards, the framework is extended in Chapter (4) to be able to benefit from feedback to increase its sample efficiency without being dependent on external support. The proposed extension is evaluated through two of the four interaction scenarios used to evaluate the original unsupervised framework. Chapter (5) proposes an additional extension of the grounding framework to utilize obtained groundings to extract goal states from natural language descriptions and simultaneously learn actions to enable natural task learning. To evaluate the extension task learning simulation is introduced and combined with the two interaction scenarios used to evaluate the hybrid grounding framework. Finally, Chapter (6) discusses and summarizes the main contributions of this thesis and presents possible future work.

2 Background and Related Work

This chapter introduces and explains key concepts and terms that are important to understand the contributions of this thesis. Additionally, it also provides a general high-level overview of related work, while work that is specifically related to one of the contributions of this thesis is discussed at the beginning of the corresponding chapters, i.e. Sections (3.2, 4.2, and 5.2).

2.1 Grounding

"Grounding" is an ambiguous term and has a variety of meanings in every day science and philosophy as outlined below.

- **Electrical Grounding** refers to the connection of a circuit to a common ground, which in most cases is the earth, to provide a common point of reference for different sources of electrical energy and for overcurrent protection [64].

- **Grounding (Earthing)** refers to the idea of connecting the human body to the electrons on the earth's surface to reduce the number of free radicals and thereby decrease acute and chronic inflammation, pain, and stress, while improving sleep [16][1].

- **Psychological grounding** refers to strategies used to cope with stress, negative emotions and traumas by establish a relationship with the ground and with the center of the human body [46].

- **Metaphyiscal grounding** refers to the idea that some entities are more fundamental than other entities and that the latter only exist when grounded through the former [84].

- **Grounding in communication** is a an idea proposed by Clark and Brennan [18] and describes the process of two or more parties establishing common ground, i.e. mutual knowledge, beliefs and assumptions, during a conversation.

[1] Due to the relatively low number of scientific studies that have investigated the benefit of earthing, there is still a lot of controversy whether the reported benefits are in fact due to earthing and not other potentially related factors.

- **Symbol or language grounding** refers to the idea that symbols, such as written or spoken words and phrases, only have meaning if they are linked to the real world [37].

The idea that all the meanings listed above have in common is the use of a reference point or entity to provide meaning to another entity. For example, a single electrical potential has no meaning unless there is another point with a different potential so that the current can flow to the lower potential to decrease the difference. Similarly, psychological grounding tries to cope with stress and negative emotions by regarding the situation that caused the negative feelings from a more detached and global perspective so that the person realizes its low significance in the bigger picture of their life. In this thesis, grounding always refers to symbol or language grounding, unless otherwise stated, which is most related to metaphysical grounding as well as grounding in communication and will be explained in more detail in the next section.

2.2 Language grounding

Language grounding or symbol grounding has been introduced in 1990 by Stevan Harnad through "The Symbol Grounding Problem" [37]. The main idea is that symbols, e.g. words and phrases, have no meaning unless they are connected to the real world. This connection does not need to be direct, which in fact one could argue is not even possible, but can be indirect by connecting symbols to CRs[2]. Each CR represents a set of invariant features inherent to all instantiations of a particular concept so that irrelevant details of specific instances are excluded. For example, the concept RED, cannot be described by a specific RGB or CMYK value that holds true for all instances of RED, instead it can only be described by a CR encoding the range of RGB or CMYK values that instances of RED can have so that ideally any instance of RED will be represented by the same CR. In practice, CRs can be obtained through different learning mechanisms like clustering or classification algorithms as described in more detail in Section (3.3.1). Important to note is that even the most accurate machine learning algorithm will obtain wrong CRs for some instances because of noisy perceptual information, e.g. due to partial occlusions or difficult light conditions.
However, not every symbol needs to have a direct link to a CR to be grounded in the real world, instead it is in many cases sufficient, if it is linked to several already grounded lower-level symbols so that the meaning of the higher-level symbol is provided by the meaning of the grounded lower-level symbols. Important to note is that this indirect grounding might be sufficient for some situations, while it is not for others. Let's take

[2]Harnad calls them "categorical representations" that are created by reducing "iconic representations" of inputs, which are obtained through an agent's sensors, to a set of invariant features that are sufficient to distinguish percepts belonging to different concepts [37].

the concept APPLE as an example, which could be indirectly grounded through the concepts of RED, GREEN and ROUND, if the latter three concepts have been grounded through their corresponding CRs. This indirect grounding would help an agent to successfully pick an instance of APPLE out of a bowl with other fruits, while it might still confuse it with a red or green ball. Another good example, which Harnad provides in [37], is that a system which has grounded HORSE and STRIPES would be able to identify instances of ZEBRA, if it has the abstract knowledge that instances of ZEBRA look like instances of HORSE combined with instances of STRIPE. However, this example is also not foolproof because there are other species that also look similar to HORSE and STRIPE, e.g. quaggas[3] and okapis.

For artificial agents, especially embodied agents like robots, language grounding is essential to enable efficient and natural communication with humans and to utilize abstract knowledge to interpret obtained perceptual information and guide the behavior of the agents. Therefore, a variety of grounding mechanisms have been proposed during the last decades employing machine learning to determine the correct links between abstract and concrete knowledge. The proposed grounding mechanisms can be split into two groups based on the employed learning paradigm, i.e. supervised or unsupervised learning. It is important to note that supervised learning refers here also to RL and is therefore different from the three paradigms introduced by Russell and Norvig [79]. The main reason to let supervised learning also encompass RL is that in both cases some form of feedback is provided, i.e. class labels or a reward signal, which is different from unsupervised learning, where no supervision is provided and the agents try to recognize patterns solely based on the input data.

Supervised learning based grounding approaches follow usually an IL approach (Section 2.4) and rely therefore on the support of another agent who already knows the correct mappings and how to efficiently support the grounding process of the learning agent. Examples are approaches that use dialog systems to ground higher-level symbols through already grounded lower-level symbols, e.g. [88], or approaches that use language games to directly teach correct groundings [96, 97]. In most cases, the tutor is a human, however, there is no reason an artificial agent could not act as a tutor as long as the agent has a sufficiently large set of grounded symbols and is able to provide the required support to the language learner.

Unsupervised approaches, which do not require support of another agent, employ usually some form of CSL (Section 2.3), which assumes that one word appears several times together with the same perceptual feature vector so that a corresponding mapping can be created [89, 91]. Example approaches are neural modeling fields [30, 31] or probabilistic model based approaches [4, 78]. The next two sections will explain CSL and IL

[3]Quaggas are unfortunately extinct but they were even more similar to horses than zebras because only their front body had stripes.

in the context of language grounding in more detail.

2.3 Cross-Situational Learning

CSL is a mechanism for word learning that is able to handle referential uncertainty by learning the meaning of words across multiple exposures. The basic idea, which has been proposed among others by Pinker [65] and Fisher et al. [26], is that the context a word is used in leads to a number of candidate meanings, i.e. mappings from words to CRs, and that the correct meaning lies at the intersection of the sets of candidate meanings. Thus, the correct mapping between a word and its corresponding CRs can only be found through repeated co-occurrences so that the learner can select the meaning which reliably reoccurs across situations [12, 90].

The original idea of CSL was developed to explain how humans learn words and several experimental studies have confirmed that humans employ CSL for word learning, if no prior knowledge of language is available. For example, Akhtar and Montague [2] conducted a study with 24 two-, three- and four-year-olds in which the children were presented with novel objects that differed in their shape and texture. During the experiment a new artificial adjective was introduced by telling the child "This is a *adjective* one", where *adjective* referred to the shape or texture of the target object. Afterwards, several other objects were shown to the child that had the same characteristic referred to by the used *adjective*. The results showed that already two-year-olds are able to use CSL to infer the meaning of initially unknown words. In a different study by Smith and Yu. [92], 28 12-month-old and 27 14-month-old infants were presented 30 times for 4s with pictures of two objects on a screen while the name of one of the objects was played via a loudspeaker. During the whole experiment the eye gaze of the infants was recorded to identify for how long they looked at each of the displayed objects and the results showed that they looked longer at the target than the other object, thus, confirming the successful use of CSL for word learning in infants.

Due to the results obtained in the experimental studies with infants and children, a variety of algorithms have been proposed to simulate CSL in humans and enable artificial agents, such as robots, to learn the meaning of words by grounding them through corresponding CRs. For example, Neural Modeling Fields Frameworks were used by several studies [28, 29, 30, 31] to ground words through corresponding CRs by utilizing co-occurrence information obtained across several situations. The main limitation of the frameworks is that they require the data of all situations to be collected in advance and provided at once to the frameworks so that they are not able to handle unseen words or CRs. Additionally, the studies only employed relatively simple scenarios to evaluate the frameworks using only single words as linguistic input and perfect synthetic perceptual data without noise.

A very common approach to ground words through noisy perceptual data are probabilistic graphical models, which have been used by many researchers in the field to ground a variety of modalities, e.g. spatial relations, actions, shapes, and colors, using many different experimental setups to investigate a diverse set of research questions [3, 4, 20, 44, 103, 104]. While the employed probabilistic models performed well in the used experimental setups and were useful to answer the proposed research questions, they required an offline training phase and were therefore neither able to ground words that were not included in the training data nor able to ground language in a continuous manner. Theoretically, the former problem can be addressed through the use of larger datasets, however, collecting large data of realistic perceptual data with annotations is non-trivial and it is in general also impossible to create a dataset that includes all existing words with all possible meanings because language is constantly changing, i.e. new words or meanings are created. Another limitation of the models is that they are not able to handle synonyms, i.e. multiple words refer to the same concept[4], which is a substantial limitation because many words are synonymous in specific contexts.
Note that according to the "Principle of Contrast" no two words refer to the exact same meaning, i.e. there are no true synonyms, and words can only be synonyms in specific contexts [17]. For example, *chocolate* and *sweets* are usually not synonymous because *sweets* has a broader meaning, however, when there is only one box of chocolate on the table and someone asks for the *chocolate* or *sweets* the words are synonymous in that context because they have same meaning, i.e. they refer to the same object.
There has been limited work to enable probabilistic grounding models to handle synonyms, e.g. [77, 78], however, the models still required an offline training phase so that the number of unseen synonyms they can handle is still limited by the data used for training.
In contrast to the studies mentioned above, the framework presented in this thesis does not require perceptual data and words to be collected in advance for offline training but is instead able to continuously learn new groundings in an online manner allowing (at least theoretically) its deployment in dynamic human-centered environments. Furthermore, the proposed framework has been evaluated through several scenarios that differ based on the used linguistic and perceptual information, which illustrated that is also able to handle synonymy and homonymy, complex sentence structures, and noisy perceptual information. A more detailed analysis of related work and in depth comparison with the unsupervised grounding component employed in the grounding framework proposed in this thesis is provided in Section (3.2).

[4] A concept can be referred to by multiple words (synonyms), while one word can refer to multiple concepts (homonyms). One concept can then be grounded through multiple CRs (homonyms), while it is not possible to have synonymous concepts, i.e. multiple concepts being grounded through the same CR. Since the framework employed in this study represents concepts only implicitly, synonyms are words that refer to the same CR and homonyms are CRs that refer to the same word.

2.4 Interactive Learning

IL in the area of language grounding refers to supervised approaches in which the language learner receives support and feedback, e.g. pointing or eye gaze, from a tutor. The latter can be a human or another artificial agent which does not only have a comprehensive repertoire of grounded language, i.e. words and phrases, but is also, at least partially, aware how the language learner works so that proper support and feedback can be provided. The main idea is that the provided support and feedback enables the learner to identify the correct mapping between a word and its corresponding CR instantly and without needing them to co-occur several times as is required for CSL (Section 2.3). A well known example is the Grounded Naming Game proposed in 2001 by Steels [96], which was originally developed to study the emergence and evolution of language but was later also used for grounding of words, e.g [93, 59].
The motivation for supervised grounding approaches is that, although children do not need any support to learn their native language, there is evidence that active support by their parents or other language proficient people simplifies word learning and therefore makes children learn faster [11]. Thus, IL based approaches are, similar to CSL based approaches, inspired by studies about how infants and young children learn words.
For example, Horst and Samuelson [39] conducted a study with 24-months old infants investigating whether they could sufficiently learn the names of several novel objects so that they were able to remember them after five minutes, which is a large enough delay to require retrieval from long-term memory. The experiments conducted in the study consisted of two main parts. First, the novel object names were taught by presenting two familiar objects with one novel object. The results showed that the children picked the target object on average more than 70% of the times, independent of whether the experimenter asked for a familiar or novel object. However, when they were presented with two previously novel objects that had been named during the first part of the experiment and one novel object they did not know the name of, they only picked objects requested by the experimenter at chance level when no feedback was provided during the first part of the experiment. In contrast, when feedback was provided in form of extensive labeling, i.e. after the child selected an object the experimenter held up the correct object and pointed to it while stating its name, e.g. "Look, this is the dog!", the number of times the correct object was selected was around 70%. Thus, feedback in the form of extensive labeling significantly increased the children's word learning performance.
In a different study, Bedford et al. [8] investigated word learning differences between 31 24-month-old infants at low and high risk for Autism Spectrum Disorder (ASD), which is a neurodevelopmental condition leading to deficits in social communication and interaction [5]. At the beginning of the experiment, the children were introduced to all

objects used during the experiment without naming them so that the novelty of objects had no influence on the obtained results. Afterwards, an experimenter showed several objects to the child, while asking to select a specific one, e.g. "Can you give me the moxi?". Once the child had chosen one of the objects, the experimenter either provided feedback by holding the correct object in front of the child and saying, e.g. "Yes/No, this is the moxi. What a nice moxi!", or just said "Thank you" without providing any feedback [8]. Finally, after the child was allowed to play for five minutes with other toys, the experimenter showed the child four times pairs of objects, of which only one had been named during the experiment, to investigate whether the child remembered which object belonged to the provided name. For two of the four target objects used during this phase, feedback had been provided during the previous phase, while for the other two no feedback had been provided. The results showed that providing feedback increased the number of words the children learned and that this increase was larger for the children that had a lower risk for ASD.

Inspired by the studies with children, supervised or interactive grounding approaches try to utilize the support of a tutor to obtain word-CR mappings in a sample efficient and highly accurate manner. The main idea is that direct teaching and feedback prevent an artificial agent from learning wrong mappings and reduce the complexity of language grounding by limiting the number of possible mappings. For example, several studies [55, 56, 87, 88] have used dialog systems to ground higher-level symbols through already grounded lower-level symbols during human-robot interactions. While the systems were able to learn new groundings in a fast and robust manner, the proposed systems only work, if a sufficiently large set of grounded lower-level symbols is available. In practice, this is difficult to obtain because it is impossible to know in advance what situations an agent will encounter after deployment in the real world and therefore which grounded lower-level symbols need to be available. Thus, the applicability of the presented grounding approach is limited and inadequate as the main or sole grounding mechanism, while it can be useful in combination with other grounding mechanisms that do not require the existence of already grounding lower-level symbols and can therefore be used to obtain them. Additionally, the systems relied on the availability of a human tutor who was aware of the set of lower-level symbols available to the dialog systems.

The need for a human tutor that knows the correct mappings also limits the applicability of the Grounded Naming Game [97], which is a grounding approach that has shown to allow artificial agents to quickly learn word-CR mappings in an interactive game like manner. The used procedure is relatively simple, i.e. an agent gets an instruction, selects the target object by pointing at it, and receives immediate feedback from a human tutor [10, 58, 59, 93, 94, 95]. The mechanism works very well and allows faster learning of new groundings than CSL based grounding approaches [9] because the feedback en-

ables the agent to substantially decrease the set of possible mappings by restricting the set of possible CRs a word can be mapped to without requiring any prior groundings. However, many of the studies that employed the Grounded Naming Game methodology only used a single word or phrase referring to a specific attribute of an object, which is completely different from real utterances used by humans that consist of many words. For example, even the utterances used to name novel objects in word learning studies with young children or infants are complete grammatically correct sentences, like "Look, this is the cheem!" [39], and not just single words, like "Cheem!". Thus, it is not clear whether the employed mechanisms would be applicable to more realistic natural language utterances. However, the biggest limitation is the need for another agent that is able and willing to support the grounding process.
Due to the efficiency and simplicity of the Grounded Naming Game methodology and the fact that it does not require any prior knowledge or previously obtained groundings, the feedback mechanism employed by the hybrid grounding framework proposed in Chapter (4) follows a similar approach. However, to circumvent the main limitation of the Grounded Naming Game methodology, i.e. that new grounding can only be obtained if an external supporting agent is available, the feedback mechanism of the hybrid grounding framework has been integrated with the unsupervised grounding mechanism proposed in Chapter (3), which also enables it to cope with incorrect feedback. A more detailed analysis of work combining unsupervised and supervised grounding and a detailed comparison with the approach used to combine unsupervised and supervised grounding in the grounding framework proposed in this thesis is provided in Section (4.2).

2.5 Task Learning

The main idea of task learning is that the number of possible tasks artificial agents deployed in human-centered environments would have to perform cannot be known in advance and depends on the specific environment, e.g. private home or restaurant, as well as the specific people who would interact with the agents, e.g. children or elderly people. Thus, agents must be able to learn in an efficient manner how to perform previously unknown tasks and without requiring support from someone who has been professionally trained to support the agents. The latter is important because the majority of people surrounding the agents and asking them to perform specific tasks will have in most cases no knowledge about how the agents work and therefore how to best teach them new tasks or efficiently support the learning of new tasks. Nevertheless, the agents should be able to utilize support, when available, to speed up the learning process and reduce the probability of mistakes. In general, except for tasks that only require the retrieval of information or the execution of functions preprogrammed into

the agents, like saying the current time or playing a song, most tasks require artificial embodied agents[5] to manipulate one or more objects, which in turn requires them to execute a series of actions to change the state or position of the target object and potentially also the positions of other objects, if they otherwise prevent the target object from being moved to the correct position [27].

Many studies have investigated how manipulation tasks can be automatically learned by artificial agents by determining the sequences of low-level micro-actions constituting the high-level macro-actions, i.e. the manipulation tasks. The exact format of micro-actions can vary significantly based on the requested task, previous knowledge of the agent, and the person that is asking the agent to perform the task. For example, micro-actions can be represented through movements of individual joints [35, 66], simple fine-grained movements of end effectors, or sophisticated and complex movements of end effectors or body parts, which allows the use of very high-level learning mechanisms, such as precise guidance through natural language instructions [88]. Representing micro-actions through movements of the end effector, requires either that inverse-kinematics have been implemented for the embodiment of the agent or that the agent has learned itself how it needs to manipulate its joints to move the end effector in the desired way. This can be seen as a form of grounding by grounding higher-level actions, i.e. moving the end effector, through lower-level actions, i.e. moving individual joints.

The used micro-action representation determines which learning approaches are feasible. For example, when micro-actions are represented through simple movements of joints or end effectors, most studies employed learning through demonstration or Reinforcement Learning (RL) [1, 36, 66, 100]. For the former, a human tutor has to demonstrate the desired action to the agent so that a policy can be derived from the recorded state-action pairs [6]. The latter, on the other hand, does not require the action to be demonstrated. Instead, it only requires a description of the goal state and discovers through trial-and-error possible policies [101]. Since the goal in this thesis is to be able to learn a task autonomously based on a description in natural language, RL is used because it only requires a description of the goal state instead of a detailed description of the required micro-actions, which would be necessary to employ some form of learning from demonstration.

[5] Embodied agents are agents, which are mobile so that they can move independently through the environment and are able to manipulate objects using a gripper or artificial hand. Thus, stationary home assistants that are currently present in many homes, do not count as embodied agents, although they are able to interact through loudspeakers and lights with the environments.

2.6 Reinforcement Learning

RL is a framework that allows artificial agents to learn how to act in a correct and optimal manner in a complex environment through the maximization of a reward signal [101]. It is inspired by how children learn without any direct form of supervision but just by observing the impact of their actions on the environment and trying to find the right sequence of actions to achieve a certain goal. In RL the environment is defined as everything that is outside of the agent. Interactions between the agent and environment happen in a loop. First the agent observes the current state of the environment and uses prior experience, i.e. observations of the effect of previously executed actions, to select an action. Afterwards, it executes the selected action. Finally, it observes the effect of the action by observing the new state of the environment and receives a reward signal. This signal specifies the long-term effect of an action and is given either by the environment or generated by the agent itself. The latter is the case, if the agent knows its goal state and is able to calculate how much the distance to the goal changed through the executed action. The overall goal of an RL agent is to obtain a policy, i.e. a function that specifies which action should be taken for all possible situations. Thereby, allowing it to maximize the cumulative reward received over time.
Typically, an RL problem is modeled as a Markov Decision Process (MDP), which can be represented as a 4-tuple $< \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R} >$, where $\mathbf{S}$ is the state space, i.e. the set of all possible states, $\mathbf{A}$ is the action space, i.e. the set of all possible actions, $\mathbf{T}$ is the transition probability function that describes the probability that action $\mathbf{a}$ in state $\mathbf{s}$ results in state $\mathbf{s}^{'}$, and $\mathbf{R}$ is a function specifying the reward received when transitioning from state $\mathbf{s}$ to $\mathbf{s}^{'}$ through the execution of action $\mathbf{a}$. In an MDP $\mathbf{s}^{'}$ depends only on $\mathbf{a}$ and $\mathbf{s}$, i.e. all previous actions and states have no effect [68].

This chapter has introduced key concepts and terms relevant for this thesis and has provided a general high-level overview of related work, while more detailed descriptions of work specifically related to one of the contributions are provided at the beginning of each contribution chapter, i.e. Sections (3.2, 4.2, and 5.2).

3 Unsupervised Open Ended Grounding of Natural Language

3.1 Motivation

Despite more than three decades of language grounding research, existing grounding frameworks are still very brittle and many research challenges that are fundamental to the deployment and acceptance of embodied agents in human-centered environments are still unsolved. This chapter focuses on challenges introduced by the dynamicity and unpredictability of human-centered environments as well as the ambiguity of natural language by addressing the following research questions.

1. Is it possible to detect AWs in an unsupervised and open-ended manner, i.e. without requiring a tutor nor an explicit offline training phase?

2. Is it possible to ground words and phrases in an unsupervised and open-ended manner while achieving similar or better grounding performance than existing state-of-the-art unsupervised grounding models that require and are limited by an offline training phase?

3. How to handle language ambiguity, like synonymy and homonymy, and enable different CRs, e.g. due to different sensors or feature extraction algorithms, to be assigned to the same concept in an unsupervised and open-ended manner?

To investigate above research questions, a novel CSL based unsupervised grounding framework is proposed that allows grounding of language in an open-ended manner, which is essential when considering deployment in dynamic and complex human-centered environments as defined in Chapter (1). The presented framework is evaluated based on its sample-efficiency and the accuracy of the obtained groundings through four scenarios that differ in the number of encountered situations, used utterances, and obtained percepts. Furthermore, the proposed framework is compared to a state-of-the-art unsupervised grounding framework, which has been used in many previous studies by different researchers.
The remainder of this chapter is structured as follows: Section (3.2) provides an overview of work directly related to the addressed research questions. The novel grounding

framework, the used scenarios and evaluation criteria, and the employed baseline framework are described in Sections (3.3–3.5). The obtained grounding results are presented and evaluated in Section (3.6). Finally, Section (3.7) concludes this chapter by answering the investigated research questions, summarizing the main contributions, and outlining both the limitations of the presented framework as well as how to overcome them.

3.2 Related Work

The motivation for unsupervised grounding approaches comes from the fact that children are able to learn the meaning of words, i.e. ground them in the real world, without any explicit teaching or supervision by already proficient language users, e.g. their parents or other adults [11]. One mechanism that has been found to allow children to ground words in an unsupervised manner and without the need for a tutor is CSL (Section 2.3), which allows to learn the meaning of words across multiple exposures while handling referential uncertainty. A number of experimental studies have confirmed that humans use CSL for word learning [2, 34, 92] and a variety of algorithms have been proposed to simulate CSL in humans and enable artificial agents, such as robots, to learn the meaning of words by grounding them through corresponding CRs.

For example, Fontanari et al. [30, 31] applied a Neural Modeling Fields Framework to a grounding scenario in which a tutor presents two objects to a learner while uttering a word that refers to one of the objects' shape or color so that the learner can infer the correct word-object mappings utilizing co-occurrence information across several situations. While the framework is overall able to infer the correct word-object mappings, it has several drawbacks. First, it requires the data of all situations to be presented at once and is thus not able to learn in an online fashion that is required in realistic scenarios in which unseen words or objects can occur at any time. Second, it is not clear whether the framework can handle noisy perceptual data because the used CRs were perfect and not created from real perceptual data. Finally, the model has only been evaluated for an extremely simple scenario with a single modality and one word utterances.

Yu and Ballard [109] and Frank et al. [32] used a machine translation model and a Bayesian model, respectively, to obtain a lexicon of word-object mappings by taking into account co-occurrence information. However, the employed models were also able to learn words when no consistent co-occurrence pattern was present by taking into account social cues, thereby, presenting a simple and indirect form of combined unsupervised and supervised grounding. The used utterances were based on real interactions between a mother and pre-verbal infant during which they were playing with different toys. The length of the utterances varied and included often several words that did not refer to the target object, like nouns referring to other objects, verbs, articles, or prepositions, however, the utterances were in general relatively short. Additionally, the studies

did not use the actual video recordings or corresponding video frames but a synthetic representation of the perceptual information so that it is not clear whether the models would be able to work with real noisy perceptual information.
Tellex et al. [103] and Dawson et al. [20] used probabilistic graphical models to ground spatial language through corresponding CRs in an offline fashion using large corpora of examples. The employed models performed well for sentences that only contained words they had encountered during training but had problems when sentences contained unknown words. This problem can be addressed through the use of larger datasets, however, they are not easy to obtain because the models require detailed annotations to learn from and it is impossible to create a dataset including all existing words with all possible meanings because language is constantly changing, i.e. new words or meanings are created. Another limitation of the models is that they are not able to handle synonyms, i.e. multiple words refer to the same concept, which is a substantial limitation because many words are synonymous in specific contexts.
Aly et al. [4] also used a probabilistic graphical model to ground spatial concepts and object categories through visual cues and geometric characteristics of objects, respectively. Interestingly, syntactic information in the form of Part-Of-Speech tags were also provided to the grounding model in addition to the words of the instruction to support the grounding process. While the model was overall able to obtain the correct groundings, it required an offline training phase and was only used to ground a small number of utterances with a relatively simple structure.
Salvi et al. [82] used a Bayesian probabilistic model to determine mappings between words and CRs of actions, object features, and effects through a human-robot interaction experiment. During the experiment the robot was executing an action while listening to a description of the performed action and its effect as provided by a human tutor, e.g. "The robot touches the yellow box, and the box is moving." [82]. The utterance was converted to a bag-of-words so that multiple word occurrences were ignored. Overall 1270 utterances were used (five different utterances for 254 different manipulations) containing 49 different words (including synonyms). Actions were discrete, while the three object features, i.e. shape, color, and size, and four effects, i.e. object velocity, robot hand velocity, relative velocity between the object and hand, and activation of the contact sensors of the hand, were continuous. Therefore, to obtain CRs for the object features and effects, X-means clustering [63] was used. In comparison to the previously described grounding studies, the study by Salvi et al. [82] considered relatively complex sentences with a relatively large number of different words as well as language ambiguity in the form of synonymy. Furthermore, it also used realistic perceptual information for the object features and effects, while it used synthetic discrete actions. However, like the previously described studies, the employed model required and offline training phase and is therefore not able to learn in a continuous manner required for complex

and dynamic human-centered environments.
Roesler et al. [77] and Roesler et al. [78] explicitly investigated grounding of synonymous words and phrases. More specifically, Roesler et al. [77] investigated the utility of different word representations for grounding of *unknown* synonyms, which are words for which at least one of their synonyms have been encountered during training while the word itself was not encountered. The probabilistic graphical model used for grounding received geometric object characteristics and action feature vectors as perceptual input and one of four different word representations, i.e. indices, Part-Of-Speech tags, semantic vectors obtained via Word2Vec[1], or Part-Of-Speech tags and semantic vectors, as linguistic input. The best grounding results were achieved when words were represented through semantic vectors in comparison to simple symbols, e.g. numbers, that encode no additional information. However, Roesler et al. [78] showed that this is only the case for *unknown* synonyms and that for *known* synonyms, i.e. synonymous words that have been encounted during training, representing words through simple symbols leads to better groundings, if the semantic information contains noise.
In contrast to the frameworks proposed in previous studies, the framework proposed in this chapter is able to continuously learn new groundings so that new words and percepts can be introduced at any time without requiring to discard previously learned groundings as is the case for iterative offline training from scratch. Furthermore, the proposed framework is evaluated through four different scenarios that differ based on their linguistic and perceptual information so that the proposed framework is evaluated for both synthetic and realistic perceptual information as well as natural language sentences of varying complexity. All employed scenarios contain synonyms, while two of the scenarios also contain homonyms, i.e. one word refers to multiple concepts, which was not the case for any of the scenarios used in the studies described above. Since the proposed framework is able to continuously learn new groundings, all synonyms are *known* synonyms so that it does not utilize any semantic or syntactic information following the results and recommendations of the previous studies [77, 78].
Some of the work presented in this chapter has also in some form been published in journals or at conferences. The first versions of the proposed framework were published in 2018/2019 [75, 76]. The first scenario was first used in 2018 and has since then be used in several studies to investigate different research questions and evaluate earlier versions of both the probabilistic baseline framework [77, 78] and the proposed framework [69, 71, 72]. Furthermore, Scenario IV and the corresponding results have been published in 2021 [74], while the employed emotion intensity and gender detection model was already published in 2020 [7]).

[1] Word2Vec uses a large corpus of plain text as input and outputs a vector space, where each distinct word is represented by a vector and the distance between two vectors corresponds to the syntactic-semantic similarity between two corresponding words [53, 54].

3.3 An Unsupervised Open Ended Grounding Framework

The novel grounding framework described in this section has been designed with three main goals in mind. First, it should work without any explicit external support because the presence of a supporting agent cannot be guaranteed. Second, it needs to perform grounding continuously and in an open-ended manner because new concepts, words, and CRs can be introduced at any time. Finally, it must be able to handle language ambiguity, i.e. synonyms and homonyms, because they are omnipresent in human conversations. Important to note is that, although the framework does not require any form of explicit external support, it still depends on the availability of interactions to obtain co-occurrence information. However, the interactions do not need to be with the learning agent employing the proposed framework, but can also be between other agents the learning agent is able to observe. Similar to other existing grounding frameworks, the proposed framework has no explicit representation of concepts that is independent of words and CRs. However, in contrast to most other frameworks, concepts are not explicitly represented through words or CRs but implicitly through the connections between them.

The proposed grounding framework consists of three parts: (1) CR creation component (Section 3.3.1), which utilizes standard clustering or classification algorithms to determine the correct CRs for encountered percepts, (2) AW detection algorithm (Section 3.3.2), which detects AWs in an unsupervised manner through CSL, and (3) Language grounding component (Section 3.3.3), which uses CSL to ground non-AW words and phrases through corresponding CRs. The individual parts of the proposed framework are illustrated below and in Figure (3.1), while they are described in detail in the following subsections.

1. **Concrete representation creation component**:
   - **Input**: Percepts.
   - **Output**: CRs.
2. **AW detection component**:
   - **Input**: Natural language instructions/descriptions, CRs, previously detected AWs, and word and CR occurrence information.
   - **Output**: Set of AWs.
3. **Language grounding component**:
   - **Input**: Natural language instructions/descriptions, CRs, and AWs.
   - **Output**: Word to CR mappings.

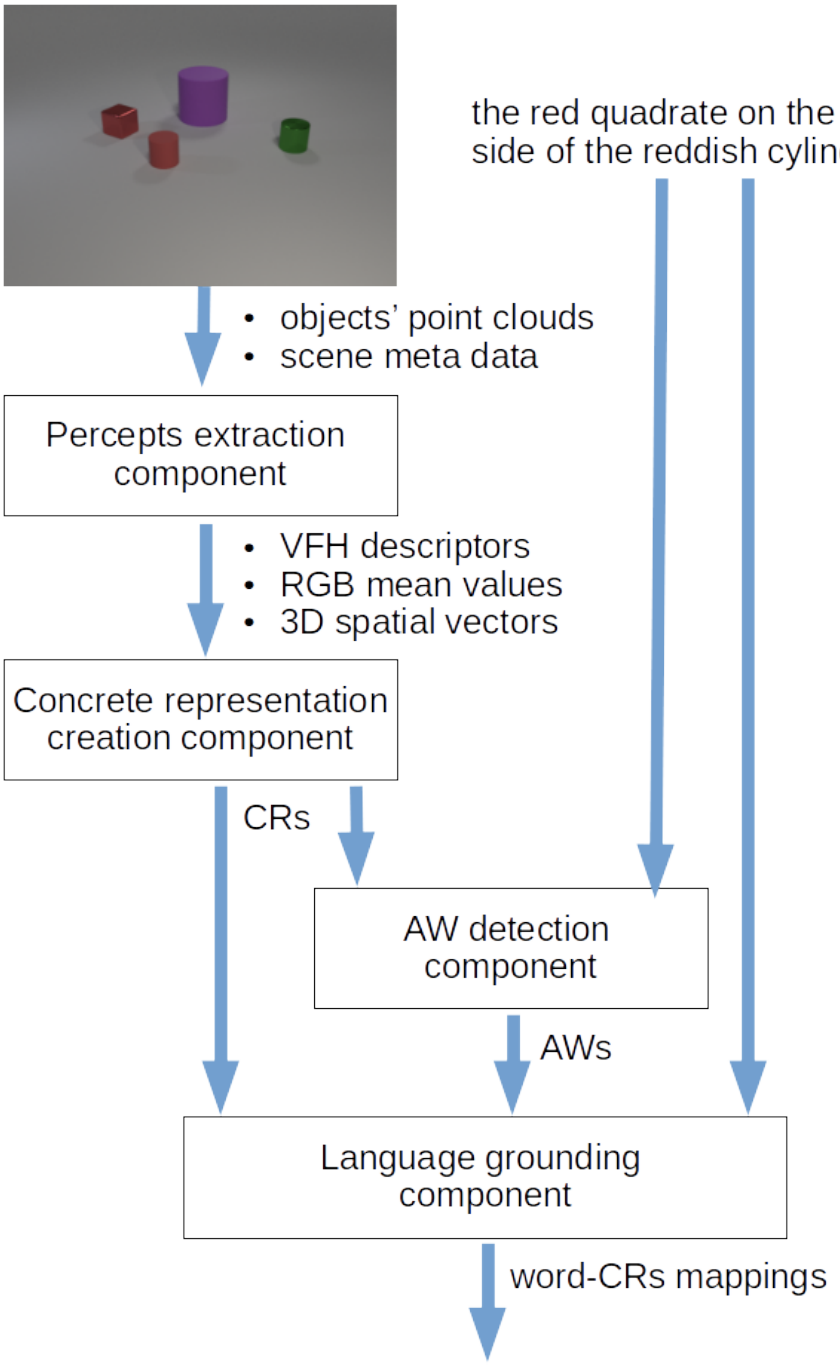


Figure 3.1: Illustration of the components of the proposed framework and the data flow for the second scenario (Section 3.5.2). First percepts, i.e. VFH descriptors, RGB mean values, and 3D spatial vectors, are extracted using the point clouds of the objects in the current scene and the meta-data generated by the scene extraction script (see Section 3.5.2 for details). Afterwards, corresponding CRs are obtained, which are then provided as input to the AW and language grounding components. Both components also take as input the natural language sentence. Finally, the language grounding component outputs the current word-CRs mappings, which take into account the current situation as well as all previously encountered situations.

3.3.1 Concrete representation creation

CRs represent sets of invariant perceptual features obtained through an agent's sensors that are sufficient to distinguish percepts belonging to different concepts by excluding irrelevant details. For illustrative purposes, let's assume that we only consider the color modality when looking at a red apple and a tomato. In this case, both are red, however, it is very unlikely that the obtained RGB values or color histograms of both objects are exactly the same. Thus, we need some mechanism that tells us reliably that these two instances belong to the same concept. Any standard clustering or classification algorithm could be used for this, i.e. take as input the perceived color percepts of both objects and output the same symbolic label, thereby, telling us that both percepts belong to the same concept. The symbolic label is then the CR of the concept RED, thus, by linking the CR to RED the latter gets grounded through all possible perceptual instances of RED.
In contrast to most grounding frameworks, the proposed framework has a separate component to obtain CRs. This has the advantage that different mechanisms, e.g. clustering or classification algorithms, can be used for different modalities and it is even possible to use multiple mechanisms in parallel for the same modality. The probabilistic model used as a baseline in this chapter (Section 3.4) has no separate component and thus relies on a specific KMeans implementation to obtain CRs in an implicit way, which limits the baseline model in several ways as described in Sections (3.6 and 3.7).

3.3.2 Auxiliary word detection

AWs are words or phrases that only exist for grammatical or linguistic reasons and have no corresponding CRs. Examples are articles, such as "a" or "the", that are used to specify definiteness or conjunctions, such as "and" or "as well as", that are used to join sentences, clauses, or phrases. Although AWs have no CRs they can still be essential for the meaning of an utterance, e.g. replacing the conjunction "neither...nor" with "both...and" reverses the meaning of the following utterance: "He neither shot the man nor threw his body into the river.".

The AW detection component of the proposed framework uses CSL to detect words that do not have corresponding CRs in an unsupervised manner. Three different mechanisms are employed for the detection of AWs, which all improve their accuracy with the number of encountered situations due to the use of CSL. The first employed AW detection mechanism is also used to detect permanent mappings that can no longer be removed once added and which are therefore different from "standard" mappings that are re-determined every time a new situation is encountered to incorporate the corresponding information. The first AW detection mechanism counts for each new situation how often each word and CR occur. If a word occurs more than three times in a given situation and all CRs only occur once, the word will be added to the Set of Auxiliary

Algorithm 1 The procedure to update sets of permanent mappings (PMS) and AWs (AWS) takes as input the lists of all words (W) and CRs (CR) of the current situation, the current PMS and AWS, and returns updated PMS and AWS.

procedure UPDATE PERMANENT MAPPINGS AND AWs(W, CR, PMS, AWS)
  $FW = \{\}, FCR = \{\}$
  for w in W **do**
    $\#w = \sum_{x \in W} 1_w(W)$
    if $\#w > 1$ **then**
      if $\#w \in FW$ **then**
        $FW(\#w) = FW(\#w) \cup \{w\}$
      else
        $FW = FW \cup \{\#w \rightarrow \{w\}\}$
  for cr in CR **do**
    $\#cr = \sum_{x \in CR} 1_{cr}(CR)$
    if $\#cr > 1$ **then**
      if $\#cr \in FCR$ **then**
        $FCR(\#cr) = FCR(\#cr) \cup \{cr\}$
      else
        $FCR = FCR \cup \{\#cr \rightarrow \{cr\}\}$
  for $\#w$ in FW **do**
    if $\#w \in FCR \wedge |FW| = |FCR| = 1 \wedge |FW(\#w)| = |FCR(\#w)| = 1$ **then**
      $PMS_w = PMS_w \cup \{FW(\#w) \rightarrow FCR(\#w)\}$
      $PMS_{cr} = PMS_{cr} \cup \{FCR(\#w) \rightarrow FW(\#w)\}$
    else if $\#w \notin FCR \wedge \#w > 3$ **then**
      $AWS = AWS \cup FW(\#w)$
  return AWS, PMS

Words (AWS). In contrast, when one word and one CR occur several times and no other word or CR occurs multiple times, they will instead be added to the set of permanent mappings (PMS) because it is a clear indication that the word is grounded by the CR. An illustration of the first AW detection mechanism is provided by Algorithm (1).

The second mechanism (Algorithm 2) identifies AWs by comparing word and CR occurrences to identify words that occurred more than any CR and at least eleven[2] times. The latter is important to avoid false detections during the first situations due to limited data. Finally, the third mechanism (Algorithm 3) detects AWs by identifying words that have a higher standard deviation across their corresponding CR occurrences in comparison to the mean standard deviation across CR occurrences for all words. Currently, there exist no mechanism to remove words from the set of AWs in case they have been incorrectly added, thus, once a word has been identified as an AW it will forever be considered as such. While it is theoretically possible that a word is incorrectly added

[2]Different thresholds have been evaluated and 11 worked well for all employed scenarios.

Algorithm 2 The second AW detection procedure takes as input the sets of word and CR occurrences (WO and CRO), and the set of detected AWs (AWS) and returns an updated AWS.

procedure AUXILIARY WORD DETECTION(WO, CRO, AWS)
  for $w, \#w$ in WO **do**
    if $\#w > max(CRO) \wedge \#w > 11$ **then**
      $AWS = AWS \cup \{w\}$
  return AWS

Algorithm 3 The third AW detection procedure takes as input the sets of previously obtained word-CR pairs ($WCRPS$), and the set of detected AWs (AWS) and returns an updated AWS.

procedure AUXILIARY WORD DETECTION($WCRPS, AWS$)
  $STD = \{\}$
  for w in $WCRPS$ **do**
    $STD = STD \cup \{w \rightarrow \{\sigma_{WCRPS(w)}\}\}$
  for w in STD **do**
    if $\overline{STD} \wedge \frac{STD(w) * WO(w)}{\overline{STD} * \overline{WO}} > 11 \wedge \overline{STD} > 14$ **then**
      $AWS = AWS \cup \{w\}$
  return AWS

to the set of AWs, the three employed detection mechanisms worked without problems for all four employed scenarios (Section 3.5), which used different words, phrases, and sentences.

However, when the framework is extended in Chapter (4) to benefit from supervision provided by an external agent, there are two cases in which non-AWs are wrongly detected as AWs, i.e. the word "cylinder" in Scenario II, when incorrect feedback is provided and the word "push" in Scenario III, independent of the correctness of the feedback. Thus, making AWs non-permanent will be necessary before considering deployment in dynamic human-centered environments and will therefore be investigated in future work.

3.3.3 Language grounding

The CSL based grounding component uses CSL to create mappings between non-AWs and phrases, which are two or more words that refer together to a concept and therefore need to be mapped as a whole to one or more CRs that represent the perceptual representation of the concept, and their corresponding CRs. Before the actual grounding procedure, the algorithm (Algorithm 4) checks whether the words of the current utterance are part of any of the phrases in the set of permanent phrases (PP), in which case it removes all words that are part of the phrase p from the set of words (W) and

Algorithm 4 The grounding procedure takes as input all words and CRs of the current situation (W and CR), the sets of previously obtained word-CR and CR-word pairs ($WCRPS$ and $CRWPS$), the set of previously detected AWs (AWS), the set of permanent phrases (PP), the sets of word and CR occurrences (WO and CRO), and the set of permanent mappings (PMS) and returns sets of grounded words and CRs (GW and GCR).

procedure GROUNDING($W, CR, WCRPS, CRWPS, AWS, PP, WO, CRO, PMS$)
  $GW = \{\}, GCR = \{\}$
  for p in PP **do**
    if $p \in W$ **then** // p is a sequence of at least two w in W
      for w in p **do**
        $W = W \setminus w$
      $W = W \cup p$
  $AWS, PMS =$ *Algorithm* 1(W, CR, PMS, AWS)
  $AWS =$ *Algorithm* 2(WO, CRO, AWS)
  $AWS =$ *Algorithm* 3$(WCRPS, AWS)$
  for aw in AWS **do**
    $W = W \setminus aw$
  for w in W **do**
    for cr in CR **do**
      if w in $WCRPS \wedge cr$ in $WCRPS(w)$ **then**
        $WCRPS(w)(cr) += 1$
        $CRWPS(cr)(w) += 1$
      else
        $WCRPS(w)(cr) = 1$
        $CRWPS(cr)(w) = 1$
  for w in $WCRPS$ **do**
    $max = 0$
    for cr in $WCRPS(w)$ **do**
      if $WCRPS(w)(cr) > max$ **then**
        $cr_{max} = cr$
        $max = WCRPS(w)(cr)$
    $GW(w) = GW(w) \cup \{cr_{max}\}$
  for cr in $CRWPS$ **do**
    $max = 0$
    for w in $CRWPS(cr)$ **do**
      if $CRWPS(cr)(w) > max$ **then**
        $w_{max} = w$
        $max = CRWPS(cr)(w)$
    $GCR(cr) = GCR(cr) \cup \{w_{max}\}$

35: **for** w in PMS_w **do**
36: $GW(w) = GW(w) \cup PMS_w(w)$
37: **for** cr in PMS_{cr} **do**
38: $GCR(cr) = GCR(cr) \cup PMS_{cr}(cr)$
39: **return** $GW \cup GCR$

adds instead p to W. Although the proposed framework has a component to detect phrases in an unsupervised manner through CSL [76], it has not been used for the experiments presented in this chapter to ensure a fair comparison with the baseline model because the latter does not have any phrase detection capabilities, thus, for the presented experiments the phrases in PP were predefined.
After all phrases have been replaced, the AW detection mechanisms described in the previous section (Section 3.3.2) are used to update the set of AWs (AWS) as well as the set of permanent mappings (PMS) in case of Algorithm (1). Afterwards, all known AWs are removed from the received natural language instruction (W) and a set of CRs is created for each word ($WCRPS$), in which each CR is saved with a number indicating how often it occurred together with that word. The same is also done for CRs, i.e. for each CR a set of words is created ($CRWPS$). Then, the highest word-CR and CR-word pairs are determined and saved to the sets of grounded words (GW) and grounded CRs (GCR), respectively.
To enable the algorithm to ground synonyms and homonyms, i.e. map multiple words to the same CR or map one word to multiple CRs, the words and CRs that were part of the highest word-CR and CR-word pairs can be used again during all future iterations. For the same reason, the algorithm also looks at both word-CR and CR-word pairs because looking only at word-CR or CR-word pairs would prevent the algorithm from handling synonyms or homonyms, respectively. Afterwards, the grounding algorithm merges the temporary mappings in GW and GCR with the permanent mappings in PMS. Finally, the sets of grounded words and CRs are merged so that the algorithm returns, in the end, a single set of mappings containing both synonyms and homonyms.

3.4 Baseline: A Probabilistic Grounding Framework

A Bayesian learning framework, which identifies AWs and grounds non-AWs and phrases through corresponding CRs, is used as a baseline in this chapter. The framework has been chosen as a baseline because similar models have previously been employed in a variety of scenarios by different researchers, e.g. [44, 103, 3, 77, 78]. In the framework, outlined in Figure (3.2), the observed state w_i represents word indices, i.e. each

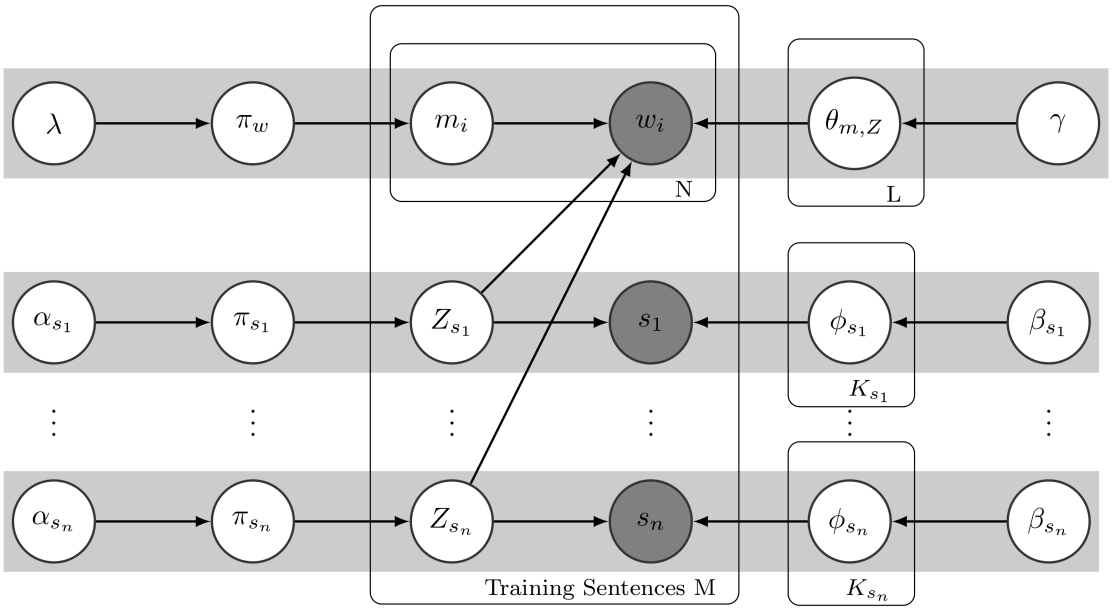


Figure 3.2: Graphical representation of the probabilistic model. Index **i** denotes the order of words, while $s_1, \ldots, s_n$ denote the observed states representing the predefined modalities.

individual word is represented by a different integer[3]. The observed states $s_1 ... s_n$ represent the different modalities, e.g. shapes, colors, and actions for the first scenario (Section 3.5.1) and emotion types, emotion intensities, and genders for the fourth scenario (Section 3.5.4), that the model is able to use for grounding of words. The fact that the available modalities need to be predefined and it is not possible to add another modality after deployment is a strong limitation of the baseline model.

Table (3.1) provides a summary of the definitions of the learning model parameters. The corresponding probability distributions, i.e. w_i, $\theta_{m,Z_{L_1}}$, $\phi_{s_{1K_1}}$, ..., $\phi_{s_{nK_n}}$, π_w, π_{s_1}, ..., π_{s_n}, m_i, Z_{s_1}, ..., Z_{s_n}, and s_1, ..., s_n, which characterize the different modalities in the graphical model, are defined in Equation (3.1), where *Cat* denotes a categorical distribution, *Dir* denotes a Dirichlet distribution, *GIW* denotes a Gaussian Inverse-Wishart distribution, and *N* denotes a multivariate Gaussian distribution. For all scenarios (Sections 3.5.1 to 3.5.4) multivariate Gaussian (*N*) and Gaussian Inverse-Wishart (*GIW*) distributions are used because the perceptual data is represented by continuous data[4].

[3]The following two example sentences, taken from the first scenario (Section 3.5.1), illustrate the representation of words through word indices: (please, **1**) (lift up, **2**) (the, **3**) (brown, **4**) (coke, **5**) and (lift up, **2**) (the, **3**) (brownish, **6**) (lemonade, **7**), where the bold numbers indicate word indices.

[4]For Scenario IV (Section 3.5.4) the perceptual data is first provided as input to deep neural networks to obtain CRs, thus, theoretically the input to the graphical model would be categorical data, however, the data is afterwards converted to one-hot encoded vectors, i.e. only one element in each vector is non-zero (hot) to indicate which category it represents (see Figure 3.6 in Section 3.5.4 for an illustration), so that the data provided to the graphical model is again continuous, in the end.

Table 3.1: Definitions of the learning parameters in the graphical model (Figure 3.2).

Parameter	Definition
λ	Hyperparameter of the distribution π_w
$\alpha_{s_1}, \ldots, \alpha_{s_n}$	Hyperparameters of the distributions $\pi_{s_1}, \ldots, \pi_{s_n}$
m_i	Modality index of each word (modality index $\in \{M_{s_1}, \ldots, M_{s_n}, \text{AW}\}$)
$Z_{s_1}, \ldots, Z_{s_n}$	Indices of percept distributions
w_i	Word indices
$s_1, \ldots, s_n$	Observed states representing the predefined modalities
γ	Hyperparameter of the distribution $\theta_{m,Z}$
$\beta_{s_1}, \ldots, \beta_{s_n}$	Hyperparameters of the distributions $\phi_{s_1}, \ldots, \phi_{s_n}$
$\theta_{m,Z}$	Word distribution over modalities

The latent variables of the Bayesian learning model are inferred using the Gibbs sampling algorithm [33] (Algorithm 5), which repeatedly samples from and updates the posterior distributions (Equation 3.2). For all scenarios, distributions were sampled for 100 iterations, after which convergence had been achieved. Due to the employed inference algorithm the baseline framework requires an offline training phase and needs to be re-trained from scratch to learn how to handle novel modalities or even previously unseen words, which makes it unsuitable for deployment in dynamic human-centered environments because it is impossible to predict all possible situations in advance.

$$
\begin{cases}
w_i & \sim & Cat(\theta_{m_i,Z_{m_i}}) & & \\
\theta_{m,Z_{L_1}} & \sim & Dir(\gamma) & , & L_1 = (1, \ldots, L) \\
\phi_{s_1 K_1} & \sim & GIW(\beta_{s_1}) & , & K_1 = (1, \ldots, K_{s_1}) \\
 & \vdots & & & \\
\phi_{s_n K_n} & \sim & GIW(\beta_{s_n}) & , & K_n = (1, \ldots, K_{s_n}) \\
\pi_w & \sim & Dir(\lambda) & & \\
\pi_{s_1} & \sim & Dir(\alpha_{s_1}) & & \\
 & \vdots & & & \\
\pi_{s_n} & \sim & Dir(\alpha_{s_n}) & & \\
m_i & \sim & Cat(\pi_w) & & \\
Z_{s_1} & \sim & Cat(\pi_{s_1}) & & \\
 & \vdots & & & \\
Z_{s_n} & \sim & Cat(\pi_{s_n}) & & \\
s_1 & \sim & N(\phi_{Z_{s_1}}) & & \\
 & \vdots & & & \\
s_n & \sim & N(\phi_{Z_{s_n}}) & &
\end{cases}
\tag{3.1}
$$

Algorithm 5 Inference of the model's latent variables. The number of iterations (*#iter*) was set to 100 for all scenarios.

procedure GIBBS SAMPLING($S_1, \ldots, S_n, w$)
    Initialization of $\theta, \phi_{s_1}, \ldots, \phi_{s_n}, \pi_w, \pi_{s_1}, \ldots, \pi_{s_n}, Z_{s_1}, \ldots, Z_{s_n}, m_i$
    for $i = 1$ to *#iter* **do**
        Equation (3.2)
    return $\theta, \phi_{s_1}, \ldots, \phi_{s_n}, \pi_w, \pi_{s_1}, \ldots, \pi_{s_n}, Z_{s_1}, \ldots, Z_{s_n}, m_i$

$$
\begin{cases}
\phi_{s_1} & \sim & P(\phi_{s_1}|s_1, \beta_{s_1}) \\
 & \vdots & \\
\phi_{s_n} & \sim & P(\phi_{s_n}|s_n, \beta_{s_n}) \\
\pi_w & \sim & P(\pi_w|\lambda, m) \\
\pi_{s_1} & \sim & P(\pi_{s_1}|\alpha_{s_1}, Z_{s_1}) \\
 & \vdots & \\
\pi_{s_n} & \sim & P(\pi_{s_n}|\alpha_{s_n}, Z_{s_n}) \\
Z_{s_1} & \sim & P(Z_{s_1}|s_1, \pi_{s_1}, w) \\
 & \vdots & \\
Z_{s_n} & \sim & P(Z_{s_n}|s_n, \pi_{s_n}, w) \\
\theta_{m,Z} & \sim & P(\theta_{m,Z}|m, Z_{s_1}, \ldots, Z_{s_n}, \gamma, w) \\
m_i & \sim & P(m_i|\theta_{m,Z}, Z_{s_1}, \ldots, Z_{s_n}, \pi_w, w_i)
\end{cases}
\tag{3.2}
$$

3.5 Experiments

The proposed framework (Section 3.3) is evaluated through four different scenarios that differ in the complexity and length of the employed utterances, the complexity and type of the used percepts, and the number of encountered situations. In the first scenario a human and robot are interacting in front of a tabletop (Figure 3.3) so that the robot grounds words through the CRs of the percepts obtained through its sensors. The main purpose of this scenario is to investigate whether the proposed framework is able to handle percepts extracted during real world interactions and whether the framework can learn from a relatively small number of interactions.

The second scenario consists of 1,000 human-agent interactions in a simulated environment that is based on the Compositional Language and Elementary Visual Reasoning (CLEVR) dataset [41]. In comparison to the first scenario, the situations in the second scenario are more complex because the employed sentences are longer, not all CRs of a particular situation are described through the corresponding utterance, and the same word or CR can appear several times in the same situation. Additionally, the situations contain not only synonyms but also homonyms because every preposition word can be

Figure 3.3: Schematic representation of the human-robot interaction in Scenario I. A robot is placed in front of a table with one object, and a human tutor provides an instruction so that the robot executes the corresponding action.

grounded through two different CRs.
The third scenario uses also simulated human-robot interactions, however, in this case the percepts are simple hot-encoded vectors without any noise to evaluate how the framework performs when the obtained CRs are perfect. Furthermore, the scenario has ten times more situations than the second scenario as well as more modalities than any of the other scenarios. Finally, the fourth scenario investigates grounding of higher level concepts, like emotion types and intensities. The main difference is that due to the high complexity of the percepts the previously employed clustering algorithm could not be used to obtain CRs of percepts, therefore, deep neural networks are instead used in Scenario IV. Table (3.2) provides a brief overview of the used scenarios that highlights their differences, while all scenarios are described in detail in the following subsections.

3.5.1 Scenario I: Sensors

The percepts used in the first scenario have been obtained during multiple interactions between a human and HSR robot[5] in front of a tabletop. During the interactions the human places one of five objects, i.e. {BOTTLE, CUP, BOX, CAR, and BOOK} (Figure 3.4a),

[5]The Human Support Robot (HSR) used for the experiment in the first scenario has a cylindrical shaped body, which can move omnidirectional, one arm with a gripper, 11 degrees of freedom, and is equipped with stereo and wide-angle cameras, a microphone, a display screen, and a variety of different sensors [106].

Table 3.2: Overview of the used percept source, modalities, perceptual representations and the number of encountered CRs, words, AWs, and situations for all four employed scenarios.

Scenario	Percept Source	Modality	Representation	# CRs	# Words	# AWs	# Situations
I (Section 3.5.1)	Sensors	Shape Color Action	VFH descriptor Histogram Kinematic feature	5 5 5	25 10 10	2	125
II (Section 3.5.2)	CLEVR	Shape Color Preposition	VFH descriptor RGB means 3D vector	3 8 4	12 16 6	1	1,000
III (Section 3.5.3)	Synthetic	Shape Color Preposition Action	One-hot vector	3 8 9 5	12 16 23 12	2	10,000
IV (Section 3.5.4)	RAVDESS	Emotion type Emotion intensity Gender	156 audio features (MFCC and PCM)	7 2 2	14 4 4	2	312

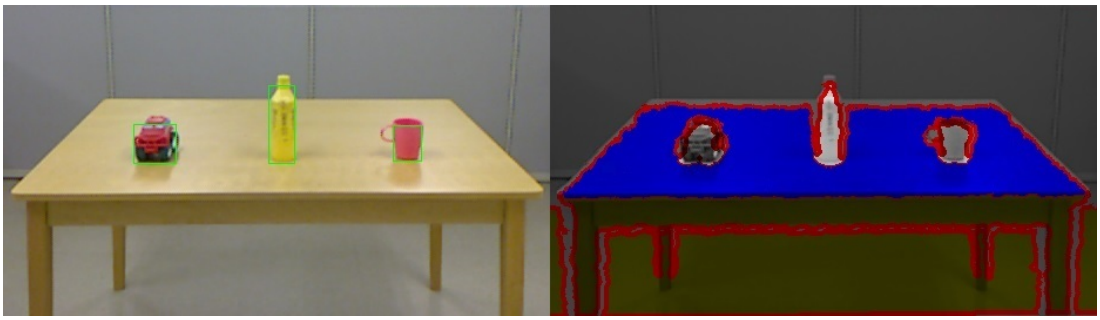
on the table and instructs the robot to perform a manipulation action on it (Figure 3.4b). Each interaction follows below procedure.

1. The human places an object on the table and the robot determines the object's geometric characteristics and color to create corresponding feature vectors (Section 3.5.1.1).

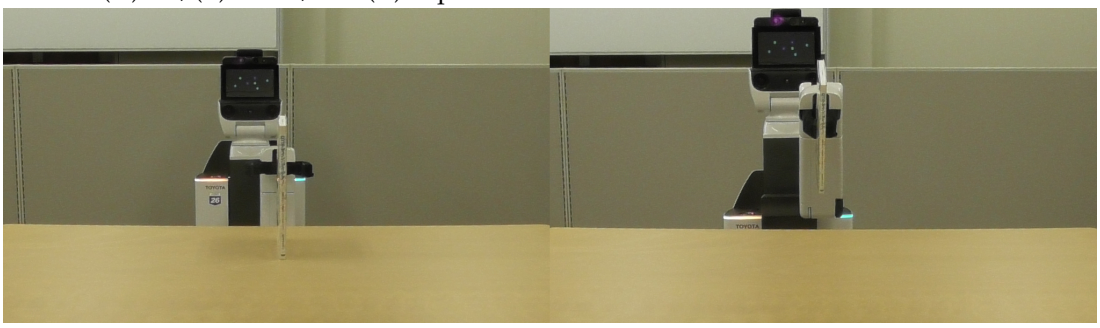
2. An instruction, which describes how to manipulate the object, is given to the robot by the human, e.g. "please lift up the red soda".

3. The human teleoperates the robot to execute the action provided through the instruction while several kinematic characteristics are recorded and converted into an action feature vector (Section 3.5.1.2).

A total of 125 interactions were performed to record perceptual information for all combinations of the employed shapes, colors, and actions. Since instruction words were selected randomly for each situation, except that words had to fit the encountered concepts, their number of occurrences in the data varies, e.g. the word "coffee" only occurs once, while the word "brown" occurs 14 times. Each sentence consists of one of the following structures: "*action* the *color shape*" or "please *action* the *color shape*", where *action*, *color*, and *shape* are substituted by one of their corresponding words (Table 3.3). Each action and color can be referred to by two different synonymous words, while each shape has five corresponding synonymous words.
Note that the used words are not actual synonyms, i.e. words that refer to the exact same meaning, but only synonyms as references to an object or action in a particular set of situations by referring to the purpose or content of an object instead of the object itself, e.g. *tea* or *coffee* instead of *cup*. However, this is sufficient for the purpose of the experiment, especially, considering that according to the "Principle of Contrast" there are



(a) Illustration of three of the five used objects and the corresponding 3D point cloud information: (A) car, (B) bottle, and (C) cup.



(b) Illustration of action *lift up* with a book as executed by the employed HSR robot in the tabletop scene.

Figure 3.4: Illustrations of the objects and actions used in Scenario I.

no words that refer to the exact same meaning, i.e. there are no "true" synonyms [17]. In the following sections the used perceptual representations are described in more detail.

3.5.1.1 3D Object Features

The object feature vectors used in the first scenario are obtained using 3D point cloud segmentation [61]. Different segmentation approaches have been investigated in the related literature. Edge based methods segment point clouds into regions by detecting their boundaries, which are characterized by points with a fast intensity change [83]. These methods are fast, but also highly sensitive to noise. Region based methods determine regions by combining neighbouring points that have similar properties [45]. They are less susceptible to noise, but are not good at determining exact region borders. Attributes based methods use predefined attributes, such as point density and vertical distribution, to cluster point clouds [24]. These methods can be very accurate and flexible, but they are often slow and the overall performance depends heavily on the quality of attributes. Graph based methods treat point clouds as a graph, where each point represents a vertex connected via edges to neighbouring points [99]. These methods can handle data with noise or uneven density, but they can not often be run in real-time.

Table 3.3: Overview of all concepts used in Scenario I with their corresponding synonyms and CR numbers (CR#) according to Figure (3.11). The actions are explained in Table (3.4).

Modality	Concept	Synonyms	CR#
Shape	BOTTLE	coca cola, soda, pepsi, coke, lemonade	1
	CUP	latte, milk, milk tea, coffee, espresso	2
	BOX	candy, chocolate, confection, sweets, dark chocolate	3
	CAR	audi, toyota, mercedes, bmw, honda	4
	BOOK	harry potter, narnia, lord of the rings, dracula, frankenstein	5
Color	YELLOW	yellow, yellowish	6
	PINK	pink, pinkish	7
	BROWN	brown, brownish	8
	RED	red, reddish	9
	WHITE	white, whitish	10
Action	LIFT UP	lift up, raise	11
	GRAB	grab, take	12
	PUSH	push, poke	13
	PULL	pull, drag	14
	MOVE	move, shift	15
Auxiliary Word	-	the	0
		please	

Table 3.4: Explanations of the actions employed in Scenario I.

Concept	Description
LIFT UP	The object will be grabbed and lifted up.
GRAB	The object will be grabbed, but not displaced.
PUSH	The object will be pushed with the closed gripper without being grabbed first.
PULL	The object will be grabbed and moved towards the robot.
MOVE	The object will be grabbed and moved away from the robot.

Model based approaches use primitive geometric shapes in order to create clusters of points with similar mathematical representations [85]. They are fast and can handle outliers, however, they are inaccurate when dealing with point clouds from different sources.

For this scenario, a model based segmentation approach is used due to its speed, reliability, and the fact that no much prior knowledge about the environment is required, such as object models and the number of regions to process [19]. The applied model detects the major plane in the environment, which is a tabletop in the conducted experiment, via the RANSAC algorithm [25], and keeps track of it in consecutive frames. Planes that are orthogonal to the major plane and touch at least one border of the image are defined as wall planes, while points that are neither part of the major nor the wall planes are voxelized and clustered into blobs. Blobs of reasonable size, i.e. neither extremely small nor large, are treated as objects. The corresponding threshold was

manually set after selecting the objects for the experiment and should be suitable for all objects of similar size. Each point cloud of a segmented object is characterized through a VFH [81] descriptor, which represents the geometry of the object taking into account the viewpoint while ignoring scale variance and color histograms representing the color of the object. Figure (3.4a) shows an example of the obtained 3D point cloud information.

3.5.1.2 Action Features

Action feature vectors are used to represent the dynamic characteristics of actions during execution through teleoperation. Overall, five different characteristics, which represent possible subactions, are recorded through the sensors of the robot [106]. The used characteristics are:

1. The distance from the actual to the lowest torso position in meters.
2. The angle of the arm flex joint in radians.
3. The angle of the wrist roll joint in radians.
4. Velocity of the base.
5. Binary state of the gripper (1: closing, 0: opening or no change).

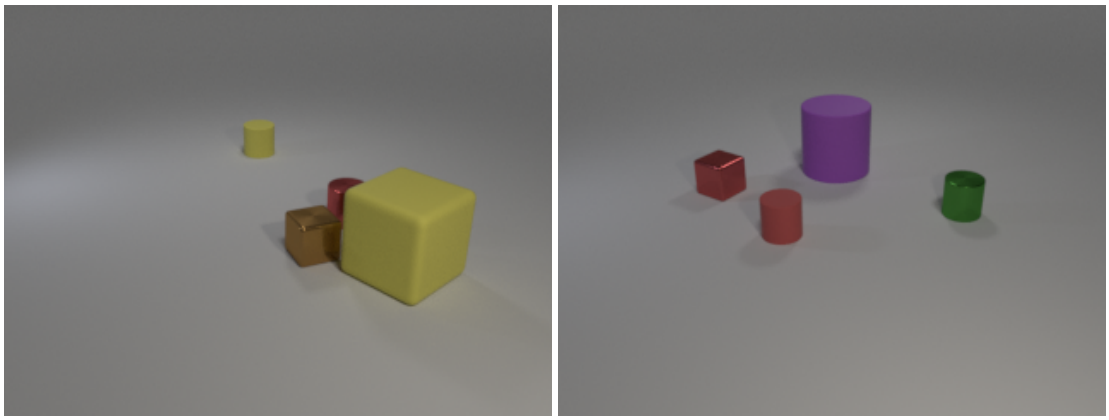
They are then combined into the following vector:

$$\begin{pmatrix} a_1^1 & \dots & a_1^5 \\ \vdots & \ddots & \vdots \\ a_6^1 & \dots & a_6^5 \end{pmatrix},$$

where a^1 represents the difference of the distances from the lowest torso position in meters, while a^2 and a^3 represent the differences in the angles of the arm flex and wrist roll joints in radians, respectively. The differences are calculated by subtracting the values at the beginning of the subaction from the values at the end of the subaction. a^4 represents the mean velocity of the base (forward/backward), and a^5 represents the binary gripper state. Each action is characterized through six manually defined subactions. Therefore, if an action consists of less than six subactions, rows with zeros are added at the end, while the duration of a subaction is not fixed because it depends on the teleoperator.

3.5.2 Scenario II: CLEVR

The situations in the second scenario are simulated using an environment based on the CLEVR dataset [41] by using a slightly modified version of the scene creation script proposed by [41] so that it also extracts and saves the point clouds of the objects from



(a) (b)

Figure 3.5: Two example scenes illustrating the used shapes and colors as well as the variation in size, material and light conditions. The corresponding sentences are: (a) "the red cylinder in front of the yellowish cylinder" and (b) "the red quadrate on the left side of the reddish cylinder".

the scene generated in Blender [6]. Every situation in the simulated environment consists of three or four objects with randomly chosen shapes, colors, materials, sizes, and positions (Figure 3.5). Additionally, every situation has different light conditions, which adds noise to the perceived color information so that the similarity of two percepts of the same color varies depending on the light conditions. Three different modalities are extracted for each situation: (1) object shapes, which are represented by VFH descriptors that were extracted from the objects' point clouds using the Point Cloud Library[7] [80] and that encode the objects' geometries and viewpoints, (2) object colors, which are represented by the mean RGB values of all object pixels, (3) preposition percepts, which are represented by 3D spatial vectors that were extracted from the meta-data generated by the scene generation script and that describe the spatial relation of the centroids of two objects.

After all perceptual information have been obtained, a random sentence describing the generated scene is created, which has the following structure: "the *color shape preposition* the *color shape*", where *color*, *shape*, and *preposition* are substituted by one of 12 shape, 16 color, and 6 preposition words or phrases (Table 3.5) to match the randomly selected target and reference objects. Most of the CRs can be referred to by several synonymous words, which are not necessarily synonyms in general but might only be synonyms for the situations encountered in this scenario, similar to the synonyms in Scenario I (Section 3.5.1), to investigate how well the proposed framework handles synonymous words and phrases. Additionally, each preposition word can be grounded through two

[6] https://www.blender.org/
[7] http://pointclouds.org/

Table 3.5: Overview of all concepts used in Scenario II with their corresponding synonyms and CR numbers (CR#) according to Figure (3.14).

Modality	Concept	Synonyms	CR#
Shape	CUBE	cube, block, hexahedron, quadrate	1
	SPHERE	sphere, ball, spheroid, pellet, globe, orb, globule	2
	CYLINDER	cylinder	3
Color	GRAY	gray, grayish	4
	RED	red, reddish	5
	BLUE	blue, blueish	6
	GREEN	green, greenish	7
	BROWN	brown, brownish	8
	PURPLE	purple, purplish	9
	CYAN	cyan, greenish-blue	10
	YELLOW	yellow, yellowish	11
Preposition	RIGHT	on the right of, on the right side of	12, 13
	FRONT	in front of	13, 15
	BEHIND	behind	12, 14
	LEFT	on the left of, on the left side of	14, 15
Auxiliary Word	-	the	0

homonymous CRs. The reason is that prepositions are not discrete because most objects need to be moved in two dimensions to reach the position of another object, therefore, if an object is in front of another object it is most of the time also on the left or right of that object. In fact, for all 1,000 situations that are part of Scenario II the centroid positions of all objects in each situation were always different in two dimensions.
Figure (3.5) illustrates this nicely because the red cylinder in Figure (3.5a) is not just in front of the yellowish cylinder as indicated by the corresponding sentence, but also on the right side of it. Similarly, the red quadrate in Figure (3.5b) is both on the left side of and behind the reddish cylinder. Thus, two different CRs can be used to ground each of the preposition words. The obtained situations are then used to simulate human-agent interactions during which the human asks the agent to select an object based on a natural language description. The employed interaction procedure is described below.

1. The human places three or four objects in front of the agent and the agent obtains the corresponding shape, color and preposition percepts.

2. The human provides a natural language description of the target object, e.g. “the red cylinder in front of the yellowish cylinder”.

3. The agent gives the obtained utterance and percepts to the employed grounding model.

Cube	1	0	0
Sphere	0	1	0
Cylinder	0	0	1

Figure 3.6: Illustration of the employed one-hot encodings for shape percepts.

3.5.3 Scenario III: Synthetic

The previous two scenarios used realistic percepts which led to mean Adjusted Rand Index (ARI) scores of 0.95 (SD: 0.06) and 0.82 (SD: 0.17) for the proposed[8] and baseline[9] framework. This shows that the clustering algorithm employed by the proposed framework is performing overall better than the one used by the baseline framework, which raises the question how much the accuracy of the clusters influences the accuracy of the groundings obtained by the two frameworks, especially, since the proposed framework clearly outperformed the baseline framework for both scenarios.
To remove the influence of the employed clustering algorithm, this scenario uses one-hot encoded vectors as percepts for all modalities (Figure 3.6) so that both clustering algorithms are able to achieve perfect clustering. Additionally, more words and modalities are used and the employed natural language sentences are also longer compared to the other two scenarios and have one of the following five structures:

- "(please) *action* the *color shape*"
- "(please) *action* the *color shape preposition*"
- "(please) *action* the *color shape preposition* the *color shape*"
- "(please) *action* the *color shape preposition* the *color shape preposition*"
- "(please) *action* the *color shape preposition* the *color shape preposition* the *color shape*"

where *action*, *color*, *shape*, and *preposition* are substituted by one of their corresponding words (Table 3.6).
The scenario consists of overall 10,000 situations and each situation contains three or four objects. The situations are used to simulate human-agent interactions during which the human asks the agent to perform an action on one of the objects. In some situations the target object or target position are described in relation to another object as illustrated by the different sentence structures listed above. The employed interaction procedure is described below.

[8]The proposed framework obtained ARI scores of 0.89, 1.0, and 0.97 for the shape, color, and action percepts in Scenario I and ARI scores of 0.85, 0.97, and 0.99 for shape, color, and preposition percepts in Scenario II.

[9]The baseline framework obtained ARI scores of 0.63, 0.98, and 0.98 for the shape, color, and action percepts in Scenario I and ARI scores of 0.63, 0.71, and 1.0 for the shape, color, and preposition percepts in Scenario II.

Table 3.6: Overview of all concepts used in Scenario III with their corresponding synonyms and CR numbers (CR#) according to Figure (3.17).

Modality	Concept	Synonyms	CR#
Shape	CUBE	cube, block, hexahedron, quadrate	1
	SPHERE	sphere, ball, spheroid, pellet, globe, orb, globule	2
	CYLINDER	cylinder	3
Color	GRAY	gray, grayish	4
	RED	red, reddish	5
	BLUE	blue, blueish	6
	GREEN	green, greenish	7
	BROWN	brown, brownish	8
	PURPLE	purple, purplish	9
	CYAN	cyan, greenish-blue	10
	YELLOW	yellow, yellowish	11
Preposition	LEFT	on the left of, on the left side of, to the left, to the left side, to the left of, to the left side of	16, 17, 18
	BEHIND	behind, backwards, toward the rear, toward the rear of	14, 15, 16
	RIGHT	on the right of, on the right side of, to the right, to the right side, to the right of, to the right side of	12, 13, 14
	FRONT	in front of, forward, toward the front, toward the front of	12, 18, 19
	ON	on top of, above, over	20
Action	LIFT UP	lift up, raise	21
	GRAB	grab, take	22
	PUSH	push, poke	23
	PULL	pull, drag	24
	MOVE	move, place, displace, put	25
Auxiliary Word	-	the	0
		please	

1. The human places three or four objects in front of the agent and the agent obtains the corresponding shape, color, and preposition percepts.
2. The human provides a natural language instruction, e.g. “move the greenish hexahedron on the right side of the grayish quadrate toward the rear”.
3. The agent performs the requested action and obtains the corresponding action percept.
4. The agent provides the obtained utterance and percepts to the employed grounding model.

3.5.4 Scenario IV: RAVDESS

The scenario described in this section differs significantly from the three previous scenarios, which were all very similar despite the use of different percepts and utterances because the agent was always told to identify or manipulate an object. The main idea of the scenario presented in this section is that the agent is listening to another person’s

Table 3.7: Overview of all concepts used in Scenario IV with their corresponding synonyms and CR numbers (CR#) according to Figure (3.20).

Modality	Concept	Synonyms	CR#
Emotion Type	HAPPINESS	happy, cheerful	1
	SADNESS	sad, sorrowful	2
	ANGER	angry, furious	3
	NEUTRAL	neutral, fine	4
	SURPRISE	surprised, startled	5
	FEAR	afraid, scared	6
	DISGUST	disgusted, appalled	7
Emotion Intensity	WEAK	slightly, lightly	8
	STRONG	very, really	9
Gender	MALE	he, man	10
	FEMALE	she, woman	11
Auxiliary word	-	the	0
		is	

voice, while receiving at the same time a description of the emotion the observed person is experiencing as well as the person's gender. Thus, the agent needs to ground words referring to different emotion types, emotion intensities, and genders through corresponding CRs extracted from raw audio features. Due to the complexity of the used audio features it is not possible to use standard clustering algorithms as for the previous three scenarios, therefore, deep neural networks are used instead (Section 3.5.4.1).
The human-agent interactions employed in the fourth scenario are simulated using The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [48], which consists of frontal face pose videos of twelve female and twelve male north American actors and actresses, who speak and sing two lexically-matched sentences while expressing six basic emotions, i.e. happiness, surprise, fear, disgust, sadness, and anger, plus calmness, and neutral. For the scenario described in this section only speaking records of the six basic emotions and neutral are used. Additionally, RAVDESS provides for each emotion a binary intensity value, i.e. normal and strong, while no intensity is provided for neutral. The dataset is partitioned into train and test sets by using the videos of the first eighteen actors (nine female, nine male) for training and the videos of the remaining six subjects (three female, three male) for testing. The training data is used to train the deep neural networks employed for the extraction of CRs (Section 3.5.4.1), while the test data is used to create 312 situations, i.e. for each person 8 videos per basic emotion and 4 videos for neutral.
Each sentence has the following structure: "(the) *gender* is (*emotion intensity*) *emotion type*", where *gender*, *emotion intensity*, and *emotion type* are replaced by one of their corresponding synonyms (Table 3.7). If *emotion type* is "neutral" or "fine", no intensity

percept and word are provided because being "very neutral" or "slightly neutral" does not make sense. Additionally, if the gender is described by a noun, i.e. "woman" or "man", it is preceded by the article "the".

3.5.4.1 Concrete representation extraction

CRs are extracted from the videos representing the situations of the scenario described in this section in two steps. First, all videos are given directly, i.e. without any pre-processing, as input to openEAR [22], which is a freely available open-source toolkit, to extract 384 speech features[10] including the minimum, maximum, and mean values for each individual speech feature. However, only the MFCC and PCM RMS features, i.e. 156 of the 384 obtained features, are provided to the classification models because they produced the best classification results based on an experimental evaluation of the available feature sets[11], i.e. each feature set and different combinations of feature sets were provided to the employed models. For the evaluation, the mean accuracies calculated across five runs for each feature set combination were compared and the model of the best-performing run was used in this study to obtain the CRs provided as input to the language grounding component.
Afterwards, the 156 audio features extracted by openEAR are used as input for three different deep learning models, i.e. one for each modality, after being normalized between zero and one. For emotion type classification, the model consists of four dense layers each followed by a dropout layer with a ratio of 0.1. The batch size and epoch size are set to 160 and 250, respectively. Rectified Linear Unit (ReLU) is used as an activation function in the first three dense layers, a Softmax function is used in the last layer, and Adam is used as an optimizer [43]. The applied model obtained an accuracy of 59.6% when classifying six basic emotions and neutral.
For emotion intensity recognition and gender recognition, the model proposed by Bagheri et al. [7] is used (Figure 3.7) with the following parameter settings: the convolutional layers are all 1D, have kernels of size 3 and use ReLU as activation functions to add non-linearity. The dropout layers are used as regularizers with a ratio of 0.1. The 1D max-pooling layers have a kernel size of four and are used to introduce sparsity in the network parameters and to learn deep feature representations. Finally, the dense layers are used with sigmoid activation functions to find the predicted binary distribution of

[10] Which features are extracted by openEAR depends on the used configuration. Three different configurations, i.e. INTERSPEECH 2009, emobase and INTERSPEECH 2013, were evaluated for this study but only the INTERSPEECH 2009 (emo-IS09) configuration [86] was used in the end because its features led to the best classification results.

[11] The available feature sets are pulse code modulation (PCM) root mean square (RMS) frame energy, mel-frequency cepstral coefficients (MFCC), PCM zero-crossing rate (ZCR), voice probability (voiceProb), and F0. Additionally, for each of the mentioned feature sets, a corresponding set with the delta coefficients is provided [86].

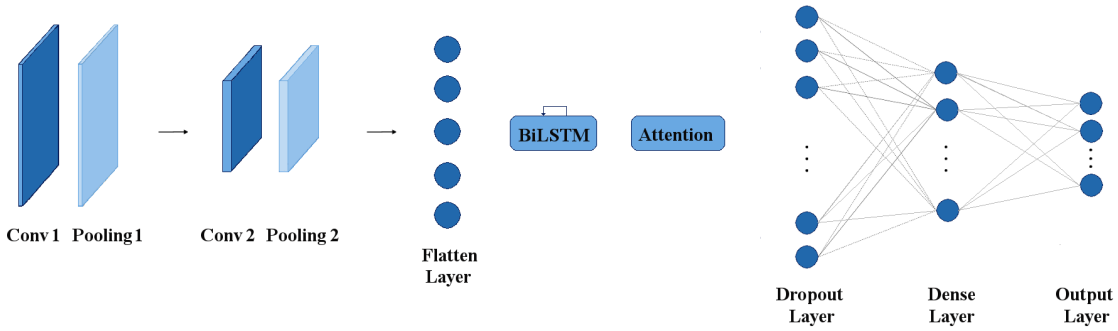


Figure 3.7: The architecture of the applied classification model for emotion intensity and gender detection in [7].

Table 3.8: Classification accuracies for all concepts and CR numbers (CR#) according to Figure (3.20).

Modality	Concept	CR#	Accuracy
Emotion Type	HAPPINESS	1	50%
	SADNESS	2	77.08%
	ANGER	3	77.08%
	NEUTRAL	4	37.5%
	SURPRISE	5	62.5%
	FEAR	6	52.08%
	DISGUST	7	56.25%
Emotion Intensity	NORMAL	8	41.66%
	STRONG	9	80.5%
Gender	MALE	10	99.35%
	FEMALE	11	78.02%

the target class. The number of epochs is 250 and the batch size is set to 128. The number of units in applied LSTM and BiLSTM networks is five. The applied model obtained an accuracy of 89.8% for gender recognition and 73.5% for emotion intensity recognition. Table (3.8) provides an overview of the classification accuracies for all CRs.

3.6 Results

In the following subsections the groundings obtained by both the proposed framework (Section 3.3) and the baseline framework (Section 3.4) for all four investigated scenarios (Section 3.5) are presented and evaluated. Since the same utterances and percepts are provided in the same sequence to both frameworks, any difference in grounding performance can only be due to the different grounding algorithms used by the frameworks. The proposed framework receives situations one after the other as if processing the data in real-time during the interaction, while the baseline framework requires all

sentences and corresponding CRs of the training situations to be provided at the same time. Therefore, two different cases are evaluated for all scenarios.
First, the case in which all situations are used for training and testing (TTS100) because the proposed framework is able to continuously learn in an online manner so that no separate training and testing phases are required. However, this represents an unrealistic case for the baseline framework because it requires an explicit offline training phase and it is very unlikely that all test situations have already been encountered during training. Therefore, for the second case, only 60% of the situations are used for training (TTS60), which is more realistic for the baseline framework, while it adds an unnecessary limitation to the proposed framework by deactivating its learning mechanism for 40% of the situations. Since situations are randomly assigned to the training and test sets, how often each word and CR occur during training and testing can vary. To minimize the influence of the used training and test sets as well as the order in which situations are presented to the proposed framework, ten different runs, i.e. sequences of situations, are used for all four scenarios.
When considering the deployability of the proposed and the baseline framework, it is important to also analyse the required computational resources. The grounding experiments have been conducted on a system with Ubuntu 16.04, i7-6920HQ CPU, octa core with 2.90 GHz each, and 32 GB RAM. However, it is important to note that both frameworks are only utilizing a single core, thus, the same processing times would be achieved on a system with a single core, if no other computationally expensive processes are running at the same time.

3.6.1 Scenario I: Sensors

This section presents the results obtained for the human-robot interaction scenario described in Section (3.5.1). The scenario serves three purposes: First, to investigate whether the proposed framework can handle real data, i.e. data obtained with the sensors of a robot. Second, to investigate the framework's ability to correctly address the ambiguity inherent to language by referring to every concept in the scenario with at least two synonymous words. Finally, to investigate the sample-efficiency of the framework by providing only a relatively small number of situations in combination with a large number of words.
Figure (3.8) shows how the mean number of correct and false mappings changes, when the proposed grounding framework (Section 3.3) encounters the employed situations one after the other. It shows that the number of false mappings is at first higher than the number of correct mappings and that both increase during the first ten situations after which the number of correct mappings is higher than the number of false mappings. The number of correct mappings increases until all situations have been encountered,

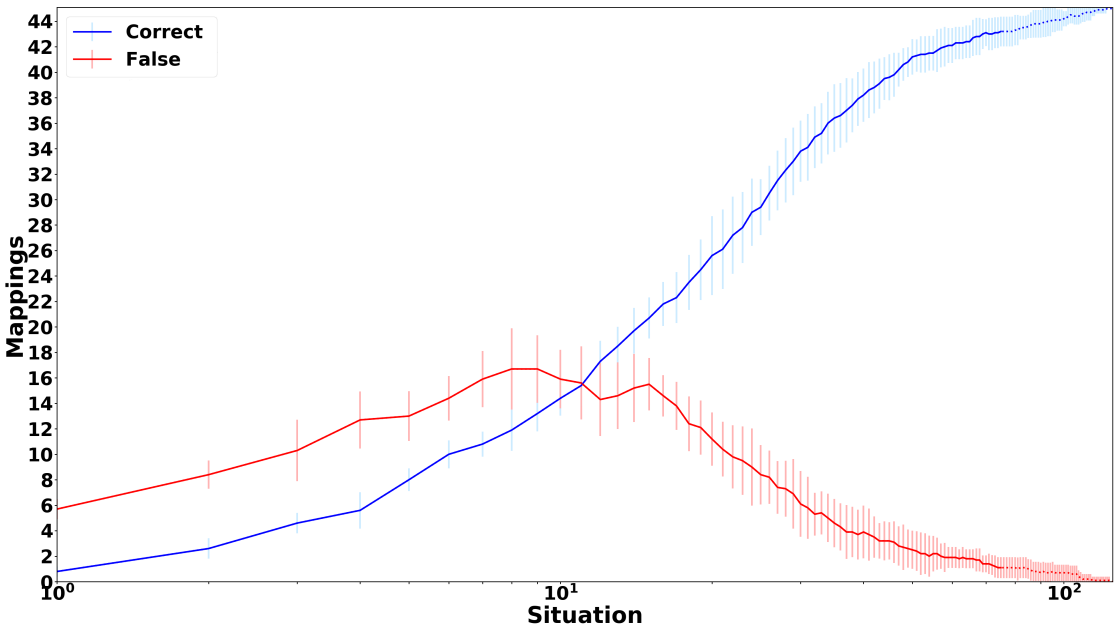


Figure 3.8: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 125 situations of Scenario I. The dotted part only occurs when all 125 situations are used for training (TTS100), otherwise, when only 75 situations are used (TTS60), the model obtains only 43 correct mappings.

while the number of false mappings decreases simultaneously. The figure shows that all 45 correct mappings are obtained, when all 125 situations are used for training, while on average only 43 correct mappings are obtained, when only 60% of the situations are used for training. In general, Figure (3.8) highlights the online grounding capability of the model, i.e. that it updates its mappings with every new encountered situation, as well as its transparency because it allows to check at any time through which CR a word is grounded at that particular moment. The collected co-occurrence information would also allow to calculate a confidence score for every mapping to understand how likely it is that a false mapping disappears or a correct mapping persists. The described transparency of the proposed framework can be helpful to understand and debug responses to instructions provided by a human, when the framework is used to control an artificial agent interacting with a human, especially when the responses are incorrect or inappropriate.

In contrast to the proposed framework, the baseline framework (Section 3.4) requires an explicit training phase so that no corresponding figure, illustrating the number of correct and false mappings, can be created. Thus, to allow a comparison between the two models, the mappings of the proposed model are extracted after 125 and 75 situations, depending on the used train/test split, i.e. TTS100 and TTS60. For TTS60, it is pos-

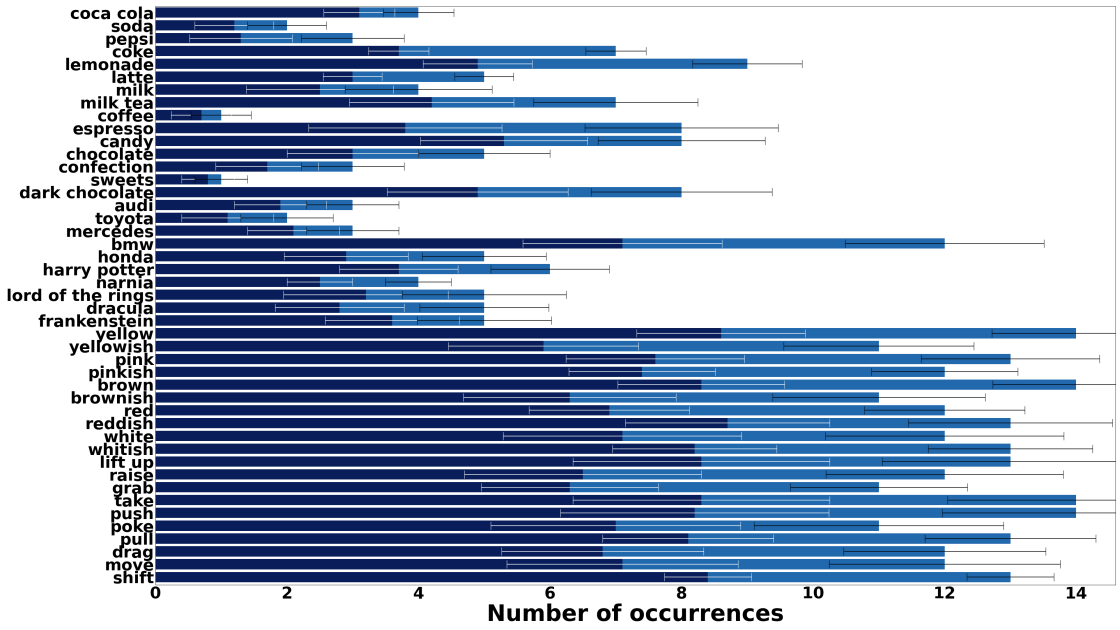


Figure 3.9: Word occurrences for all words except AWs encountered in Scenario I. The dark blue part of the bars shows the mean number of occurrences during training and the bright blue part the mean number of occurrences during testing.

sible that some words never occur during training or only a limited number of times. For example, the words *coffee* and *sweets* exist each only once in the dataset and are thus only present during training or testing, but not both, while the words *yellow*, *brown*, *take*, and *push* occur 14 times in the dataset and are thus encountered multiple times during training and testing (Figure 3.9). If a word does not occur during training, the proposed model is not able to obtain a corresponding mapping so that the word is not grounded through any CR as shown in Figure (3.11f). How often a word is encountered during training also affects the grounding performance of the baseline model, which is also not able to ground the words *coffee* and *sweets* correctly, when only 60% of the situations were used for training (Figure 3.11g).

Figure (3.10) shows that the proposed model achieves perfect grounding, when the same situations are provided for training and testing, which confirms that it is able to obtain all correct mappings as shown in Figure (3.8). However, if only 60% of the situations are used for training and the remaining 40% for testing, the grounding accuracy drops for both models. For the proposed model only the accuracy for shapes decreases to about 93.5%, while all color and action groundings as well as AWs are still correct. For the baseline model the largest drop in accuracy is seen for shapes, from more than 95% to less than 2%. The reason might be that every shape word has 5 synonyms, thus, if words would be equally distributed among all situations and specifically among the

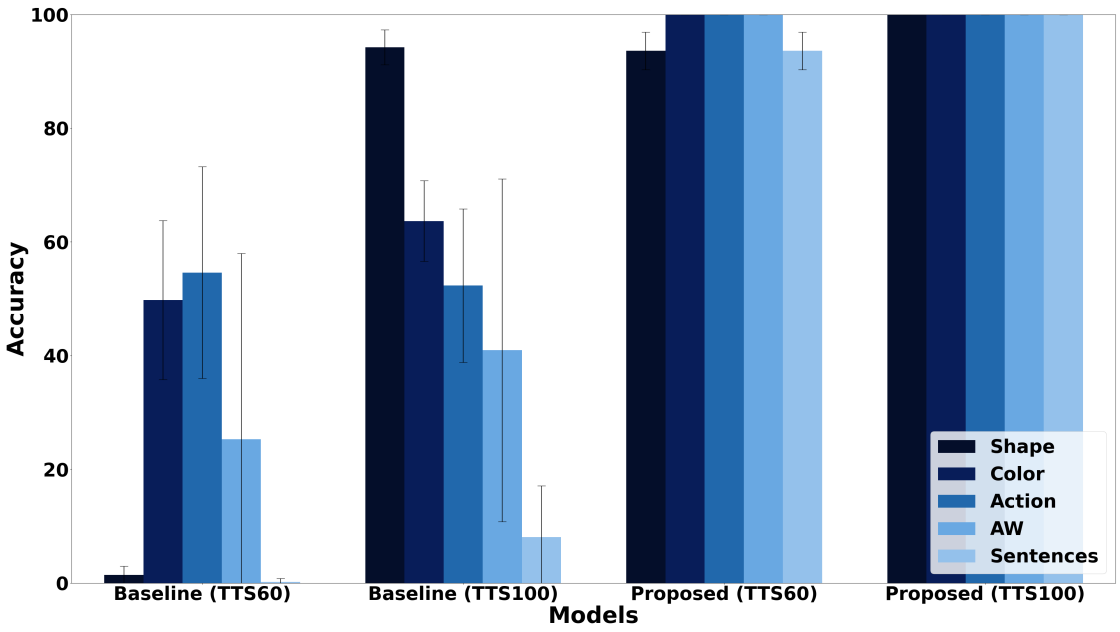
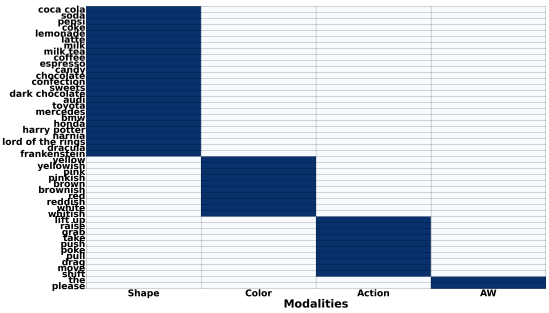


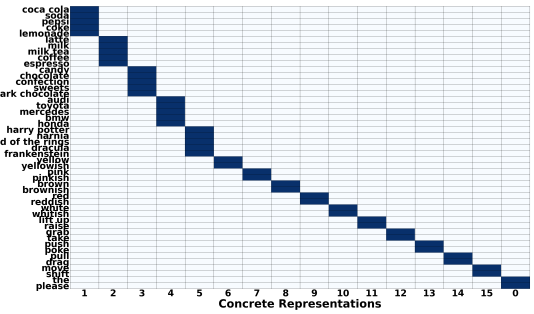
Figure 3.10: Mean grounding accuracy results and corresponding standard deviations for both grounding models and all modalities of Scenario I as well as both train/test splits. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

training and test sets, the decrease might not be as sharp.

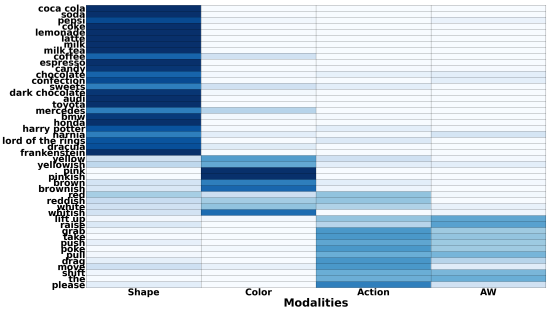
The confusion matrices in Figure (3.11) show how often each word was grounded through which modality and CR. Figures (3.11a and 3.11b) confirm that the proposed framework was able to ground all words through the corresponding CRs when the learning mechanism was enabled for all situations (TTS100). In contrast, when the learning mechanism was only enabled for 60% of the situations (TTS60) there was light confusion for two of the five book names, however, only for one of them, i.e. the word "narnia", the confusion was across modalities, while the confusion for both of them was mostly between CRs of shapes, i.e. they were partially mapped to the CR of BOX (Figures 3.11e and 3.11f). For the baseline model, Figures (3.11c and 3.11g) show that action names are often marked as AWs, while three of the ten color names were often seen as actions or shapes. Interesting is also that for TTS100 shapes are usually assigned to the CRs of the shape modality, while for TTS60 they are mostly mapped to CRs of actions or marked as AWs. When looking at the confusion matrices for CRs it becomes clear that many of the words which are mapped to the correct modality are mapped to the wrong CR, thereby, illustrating how important it is to look at the exact mappings because words must in the end be grounded through specific CRs and not just the correct corresponding modalities.



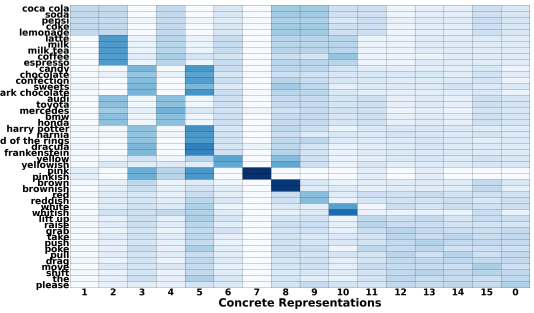
(a) Confusion matrix for the proposed model and TTS100.



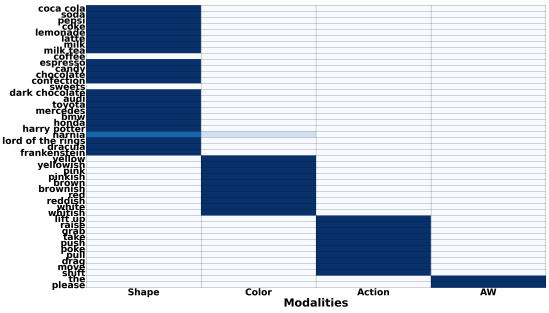
(b) Confusion matrix for the proposed model and TTS100.



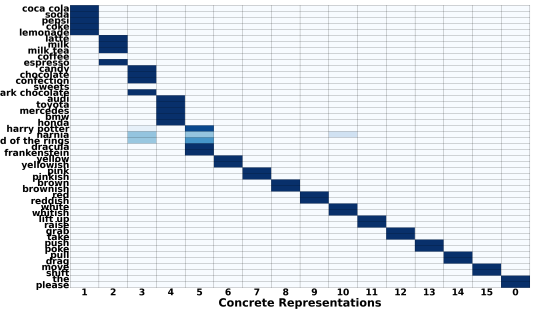
(c) Confusion matrix for the baseline model and TTS100.



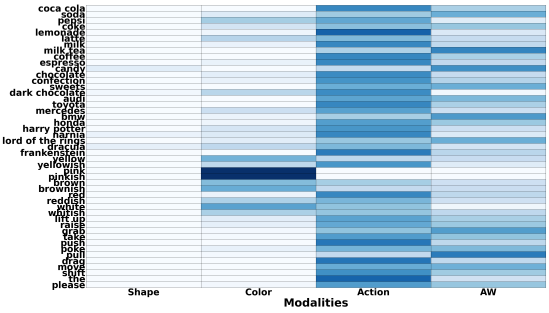
(d) Confusion matrix for the baseline model and TTS100.



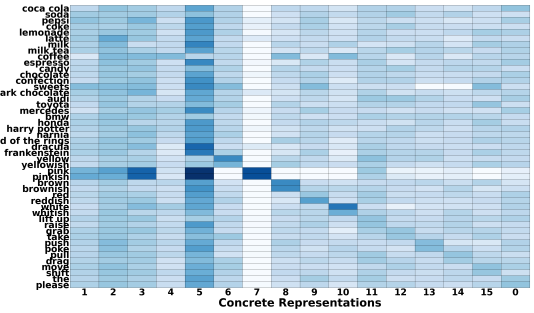
(e) Confusion matrix for the proposed model and TTS60.



(f) Confusion matrix for the proposed model and TTS60.



(g) Confusion matrix for the baseline model and TTS60.



(h) Confusion matrix for the baseline model and TTS60.

Figure 3.11: Confusion matrices showing how often each word of Scenario I was grounded through which modality and CR.

The average time it took the proposed framework to process a new situation and update its mappings was 8.32ms, while the inference time was only 60.5 µs. In contrast, one Gibbs sampling iteration of the baseline model took 23s. Since 100 iterations were used, average training time for the baseline model was about 38.5 minutes for all 125 situations, while the inference time was on average 1.15s for each situation. This means that it took the baseline model about 2,220 times longer than the proposed framework to process all 125 situations, while the proposed framework was able to do inference 19,000 times faster than the baseline model. The main reason is that Gibbs sampling becomes very slow for high-dimensional vectors like the 308 dimensional VFH descriptors used to represent the shapes of the objects employed in Scenario I. The timing analysis shows that the proposed framework is able to update its mappings during real-time human-agent interactions, while the baseline model is not able to update its groundings during interactions and even the time it takes the baseline model to do inference might be too large for dynamically changing environments.
Overall, the evaluation of the results for the first scenario shows that the proposed framework is able to handle real world data and synonyms by learning the correct mappings after just 125 situations, which is a sign that the framework is also relatively sample-efficient, when considering the words and percepts used in the the first scenario. Furthermore, the proposed framework outperforms the baseline based on its AW detection and grounding accuracy as well as its ability to obtain new mappings during interactions as illustrated by the timing analysis. Interestingly, the performance difference is larger for TTS60, although this case is artificially harming the proposed framework by preventing it to learn from all encountered situations. Finally, the proposed framework is more transparent because mappings in the proposed framework are represented explicitly and can be retrieved after every situation, which becomes important when robots are interacting with humans in complex and unrestricted environments, especially if some actions of the robots can cause harm to humans.
However, the employed scenario has also several limitations. First, the used sentences and situations are very simple because every situation contains only a single object. Second, every CR present in a situation has a corresponding word in the utterance, which is not the case in the real world where an agent would usually perceive more CRs than an utterance provided by another agent would refer to. Finally, the scenario contains many synonyms but no homonyms, although many words can refer to multiple concepts depending on the context they are used in.

3.6.2 Scenario II: CLEVR

In this section the results for the CLEVR based scenario (Section 3.5.2) are described. The scenario is more complex and difficult than the first scenario because it has longer

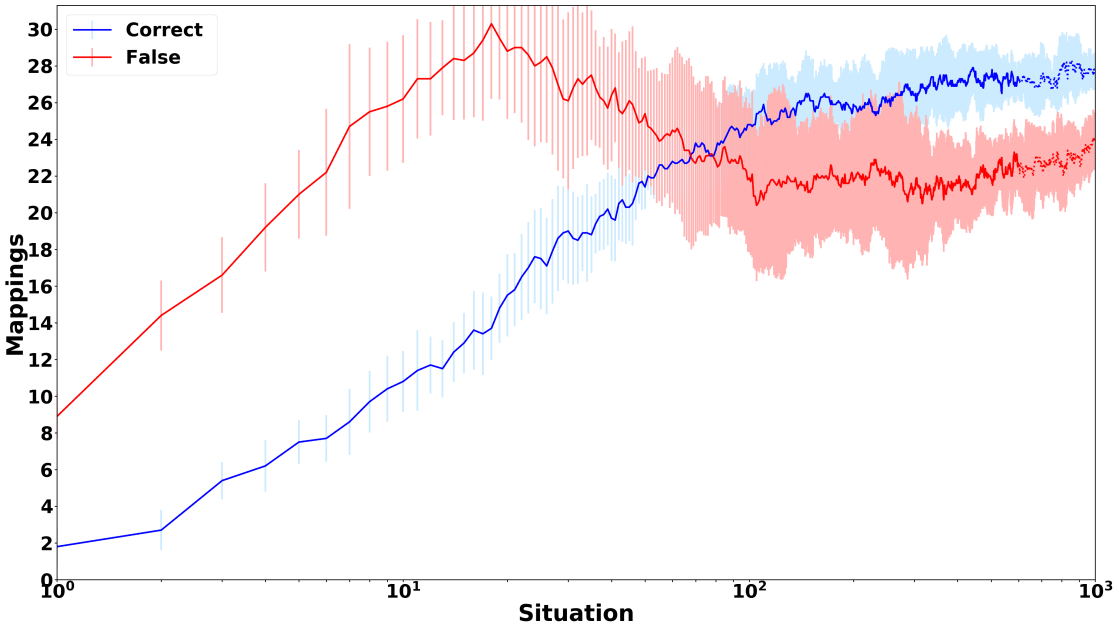


Figure 3.12: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 1,000 situations of Scenario II. The dotted part only occurs, when all situations are used for training (TTS100).

and more complex sentences, situations with several objects and therefore multiple percepts for each modality, and homonyms. As a result, the CSL based grounding algorithm of the proposed framework is only able to successfully ground about 28 of the 34 words included in the second scenario through their corresponding CRs. During the first situations most created mappings are false because the algorithm has not much data available. After around 67 situations the number of correct mappings equals for the first time the number of false mappings (Figure 3.12). 20 situations later, i.e. after about 88 situations, the number of correct mappings is for a brief moment one last time smaller than the number of false mappings. After 600 situations the number of correct mappings is 27, while after 1,000 situations it is 28. In contrast to the first scenario for which the number of incorrect mappings was zero at the end, it is even after 1,000 situations still relatively high with 21 incorrect mappings.

Important to note is that the overall number of mappings, i.e. correct and false mappings combined, is 47 after 1,000 situations, which are nine mappings more than the number of possible correct mappings[12]. The reason is that the proposed framework allows all words and CRs to be part of multiple mappings to address synonymy, i.e. multiple words are grounded through the same CR, and homonymy, i.e. one word is grounded through multiple CRs. Nevertheless, the influence of the 21 incorrect map-

[12]The overall number of possible correct mappings is 38 because there are 34 words and four of them are preposition words that can be mapped to two homonymous CRs.

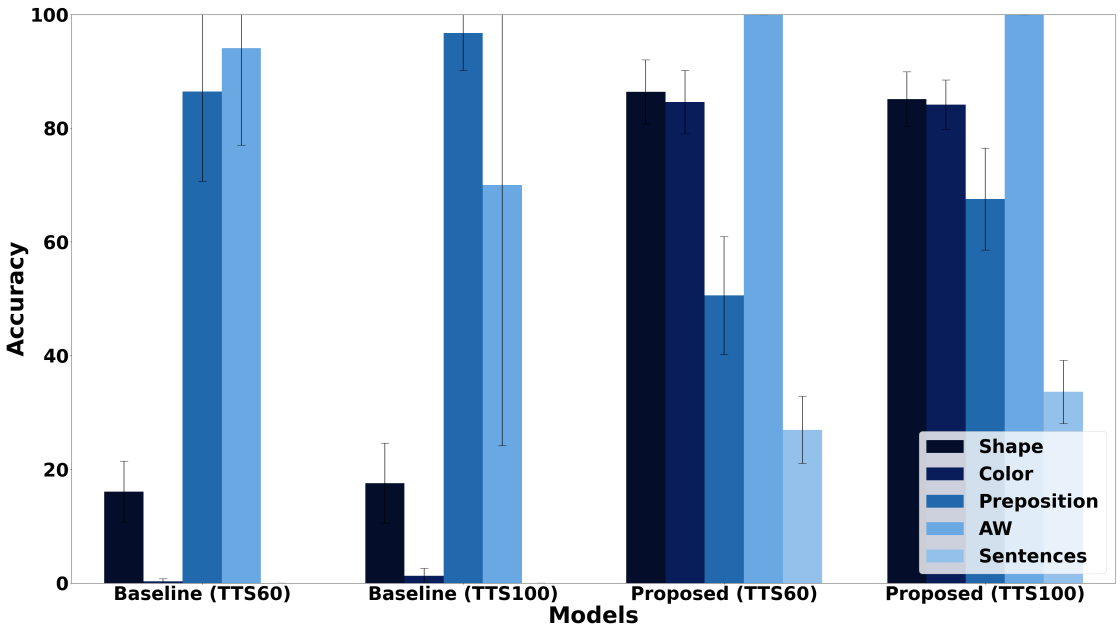
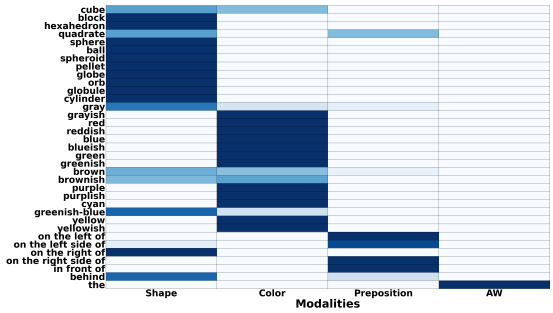


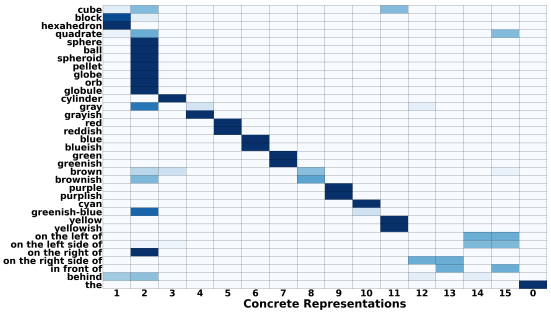
Figure 3.13: Mean grounding accuracy results and corresponding standard deviations for both grounding models and all modalities of Scenario II as well as both train/test splits. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

pings is only marginal in comparison to the influence of the 10 missing correct mappings because a missing mapping means that the agent will in general not be able to identify the CR a word belongs to or to use the correct word for a specific CR, while an incorrect mapping will only have an influence in specific situations where the correct mapping cannot be applied.

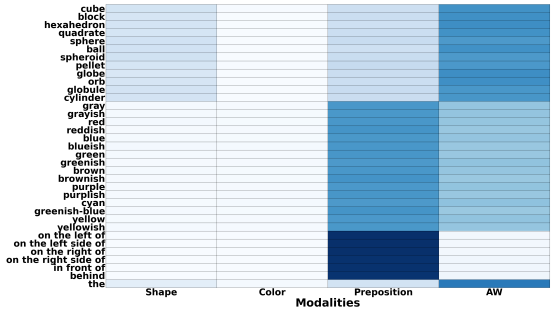
Figure (3.13) confirms this because the accuracy of the shape and color groundings obtained by the proposed framework is with more than 85% relatively high, while the accuracy of the preposition groundings is lower with only 50% for TTS60 and about 70% for TTS100 because the majority of the missing mappings are mappings for preposition words. In contrast, the baseline model achieves the highest accuracy of about 90% for prepositions, while the accuracies for shapes and colors are very low with less than 20% and 2%, respectively. Additionally, the baseline also achieves mean accuracies for AWs of about 95% and 70% for TTS60 and TTS100.
The reason for the large difference between the modalities becomes clear when looking at Figures (3.14c and 3.14g), which shows that the high accuracy for prepositions and AWs is due to the model mapping nearly all color and preposition words to preposition CRs, while all shape words are mostly mapped to the AW "the" and only lightly to shape CRs. The latter results in an accuracy of nearly 20% for shapes in comparison to less than 1% for actions.



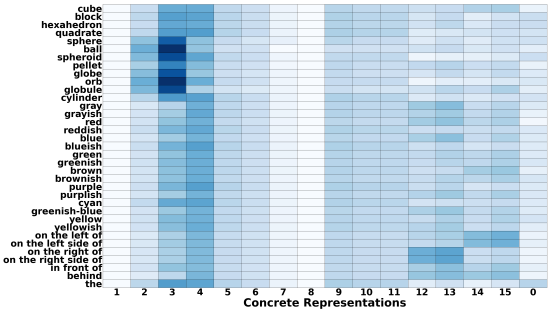
(a) Confusion matrix for the proposed model and TTS100.



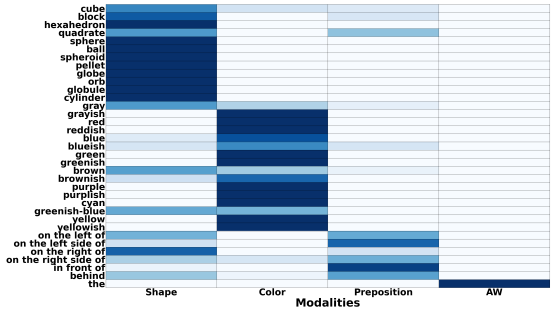
(b) Confusion matrix for the proposed model and TTS100.



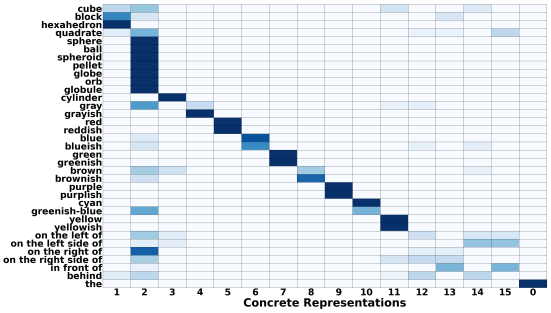
(c) Confusion matrix for the baseline model and TTS100.



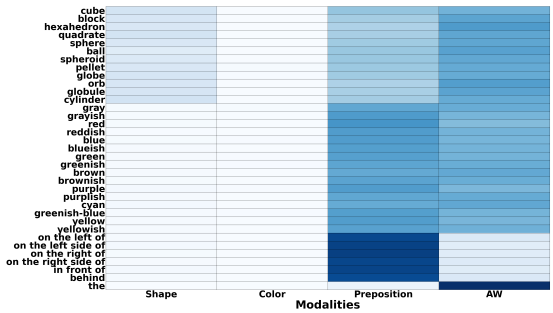
(d) Confusion matrix for the baseline model and TTS100.



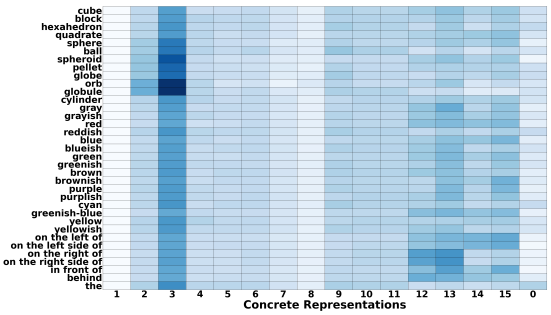
(e) Confusion matrix for the proposed model and TTS60.



(f) Confusion matrix for the proposed model and TTS60.



(g) Confusion matrix for the baseline model and TTS60.



(h) Confusion matrix for the baseline model and TTS60.

Figure 3.14: Confusion matrices showing how often each word of Scenario II was grounded through which modality and CR.

Figures (3.14a and 3.14e) show that the amount of confusion for shapes and colors is the same for TTS60 and TTS100, however, which words are incorrectly grounded and to which degree changes in some cases. For example, for TTS60 "blue" and "blueish" are partially grounded through shapes and prepositions in case of "blueish", while they are completely grounded through colors for TTS100, i.e. after all 1,000 situations have been encountered. In case of prepositions, the confusion is higher for TTS60, which is consistent with the accuracies shown in Figure (3.13), i.e. the accuracy for preposition is 20% higher for TTS100. Figures (3.14b and 3.14f) show that most confusion is across modalities independent of the number of situations used for training. More specifically, most confusion is due to words being incorrectly grounded through CR 2 of the concept SPHERE including two of the words referring to CUBE, i.e. "cube" and "quadrate", which are also the only cases of intra-modality confusion for TTS100. In contrast, for TTS60 there is also light intra-modality confusion for the phrase "on the left of".
Both the average time it took the proposed framework to process a new situation and update its mappings, and the inference time were with 18ms and 146 µs about twice as high as the times obtained for Scenario I, which is not surprising because Scenario II has eight times more situations, while the CRs are very similar. In contrast, one Gibbs sampling iteration of the baseline model took more than 5 minutes. Since 100 iterations were used, the average training time for the baseline model was more than 9 hours for all 1,000 situations, while the inference time was on average 3.39s for each situation. These results confirm the results obtained for the first scenario, i.e. the proposed framework would be able to update its mappings in real-time during human-agent interactions, while the baseline model requires too much time for training and inference.
Overall, the results show that the proposed framework is able to handle more complex situations with longer and more complex sentences describing not just the target but also a reference object and the spatial relation between the two objects, multiple objects so that not all CRs have a corresponding word in the provided natural language descriptions, and homonyms. Furthermore, the results also show that the higher complexity and difficulty of the scenario widened the performance gap between the proposed framework and the baseline model both in terms of the accuracy of the obtained groundings as well as the aquisition and inference speed.

3.6.3 Scenario III: Synthetic

The results presented in Sections (3.6.1 and 3.6.2) for the previous two scenarios show that the proposed framework achieves more accurate groundings than the baseline model, especially for the more complex second scenario. Since both frameworks use different clustering algorithm to obtain CRs of percepts and the clusters obtained by the proposed framework using DBSCAN were more accurate than the clusters obtained

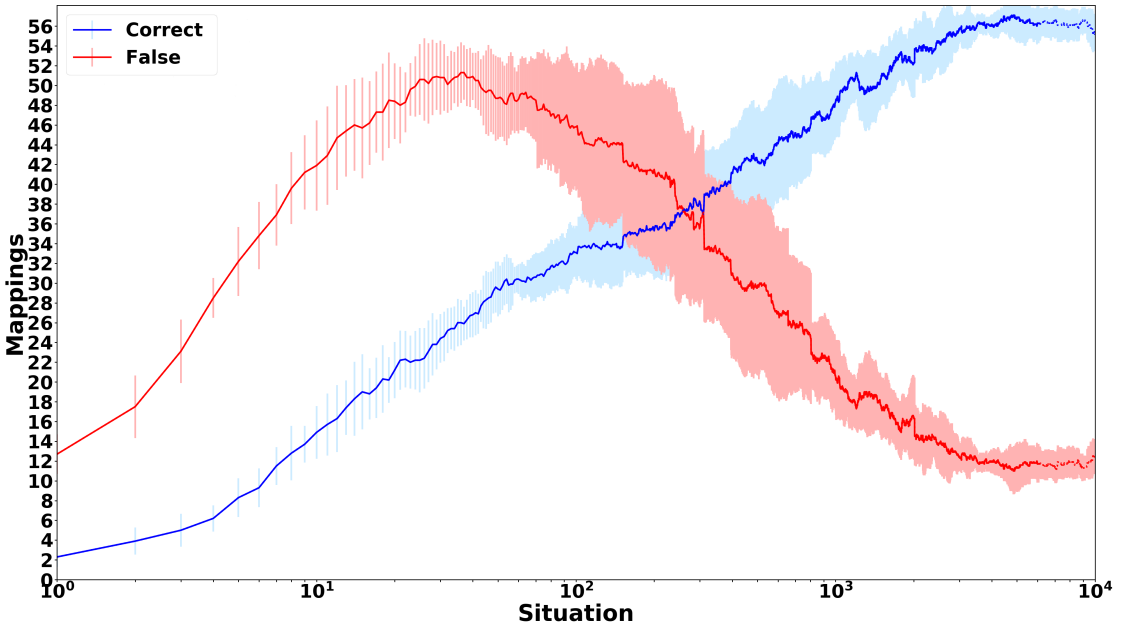


Figure 3.15: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 10,000 situations of Scenario III. The dotted part only occurs, when all situations are used for training (TTS100). Due to the large number of situations the number of correct mappings is the same in both cases.

by the baseline model using KMeans[13], the question arises whether the difference in clustering accuracy contributes to the difference in grounding accuracy. To investigate this question, the scenario investigated in this section represents all percepts through one-hot encoded vectors as described in Section (3.5.3) so that perfect CRs are obtained independent of the employed clustering algorithm, thereby negating the influence of CR accuracy on the grounding accuracy. Additionally, the scenario uses more complex sentences to investigate whether the proposed framework is able to handle a set of more realistic and diverse sentences.

The proposed framework is able to obtain about 57 of the 103 possible mappings[14] after about 5,000 situations, while there are also about 11 incorrect mappings at that time (Figure 3.15). At the beginning, the number of incorrect mappings is significantly higher than the number of correct mappings due to the large number of words and CRs. After about 35 situations the number of incorrect mappings reaches its peak with about 51 incorrect mappings in contrast to only 26 correct mappings. Afterwards, the number of

[13]The clusters obtained by DBSCAN and KMeans for all modalities of Scenarios I and II achieved mean ARI scores of 0.95 (SD: 0.06) and 0.82 (SD: 0.17), respectively.

[14]There are 12, 16, 63, and 12 possible shape, color, preposition, and action mappings, respectively. The large number of possible preposition mappings is due to the fact that four of the five preposition concepts have three CRs and that the scenario also includes 23 preposition words.

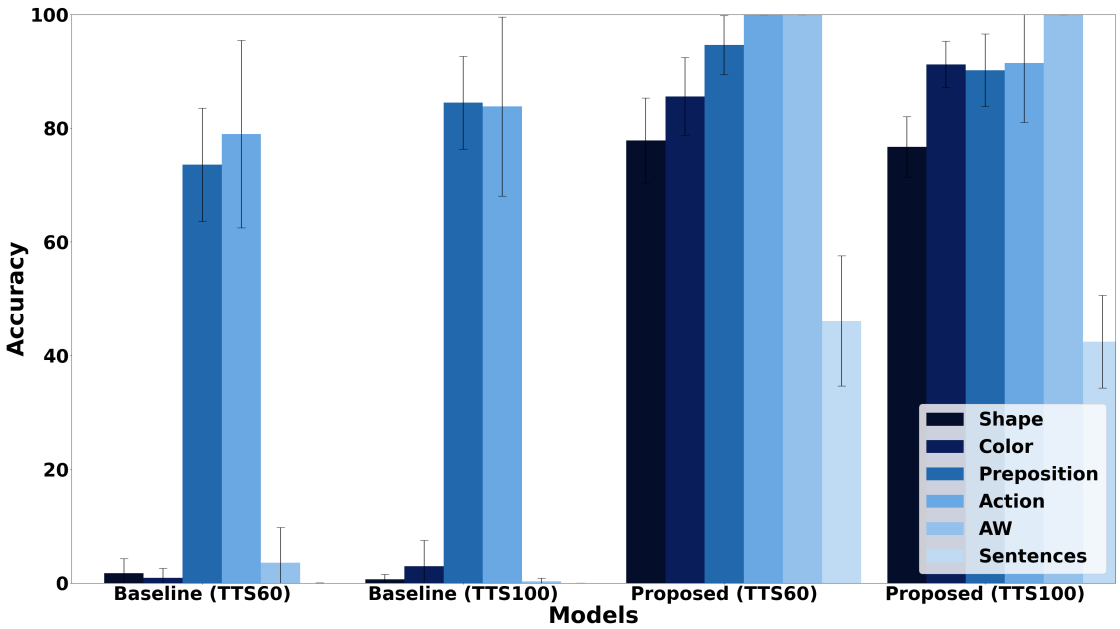


Figure 3.16: Mean grounding accuracy results and corresponding standard deviations for both grounding models and all modalities of Scenario III as well as both train/test splits. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

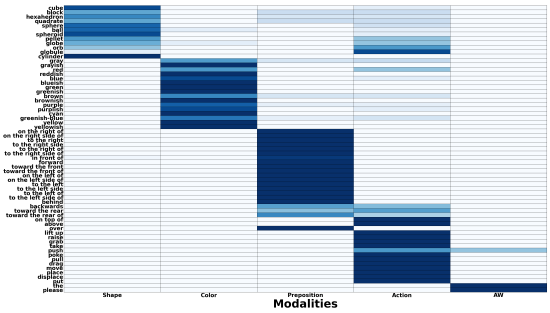
incorrect mappings decreases for about 5,000 situations to 11 incorrect mappings, while the number of correct mappings is continuously increasing to 57 correct mappings. Afterwards, it decreases to 56 correct mappings over several hundred situations and stays at that number for nearly 4,000 situations after which it briefly increases to nearly 57 correct mappings before decreasing again to this time 55 correct mappings. This change shows how the proposed framework constantly updates its mappings based on the information in new situations.

Figure (3.16) shows that the proposed framework is able to identify all AWs and for TTS60 ground also all action words correctly. Interestingly, for TTS100 the accuracy of action groundings is with more than 90% lower than for TTS60, while the accuracies for shapes and colors increase when all situations are used for training. This highlights that more situations do not necessarily lead to more accurate groundings depending on the quality of the situations. Independent of the number of situations used for training, the accuracies for colors and prepositions is above 90%, while the accuracy for shapes is a bit lower with about 75%. That there is only a minor difference in terms of grounding accuracy between TTS60 and TTS100 is not surprising because the scenario has overall 10,000 situations so that even for TTS60 the model is encountering 6,000 situations during training.

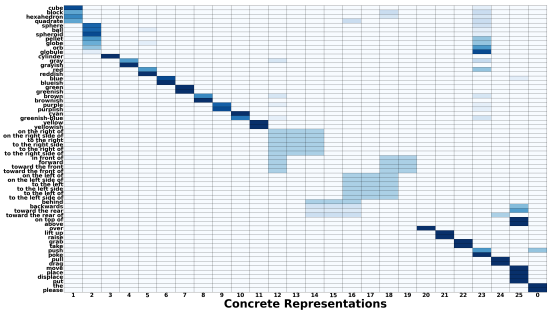
The same is true for the baseline model, i.e. there is only a slight increase in the ac-

curacy for colors, prepositions, and actions when all situations are used for training, while the accuracy for shapes lightly decreases. However, the groundings obtained by the baseline model are in general less accurate, i.e. while the accuracies for prepositions and actions are around 85%, the accuracies for shapes, colors, and AWs are below 5%. Figures (3.17c and 3.17g) illustrate that this is because the baseline model is mapping most shape and color words to the CRs of prepositions, while most action words are grounded through CRs of actions. For the proposed framework the highest confusion is shown for shapes, which are partially mapped to action CRs independent of the number of situations during which the framework was allowed to update its mappings. There also exist some confusion for colors and prepositions. The latter only exist for TTS100, which confirms the grounding accuracy for prepositions shown in Figure (3.16). In general, all confusion for the proposed model is across modalities, while there is no intra-modality confusion. Most confusion is actually with CR 23 of the concept PUSH. Figures (3.17b and 3.17f) illustrate nicely that the proposed framework is able to handle homonyms because all preposition words are grounded through the three correct corresponding CRs.

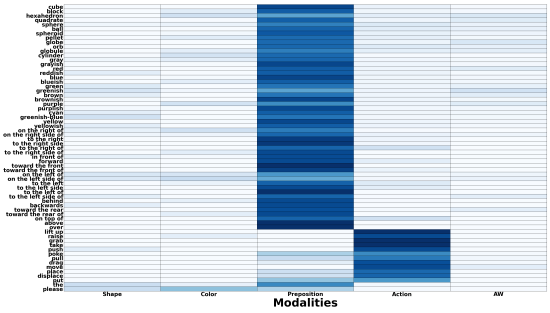
The average time it took the proposed framework to process a new situation was 67ms, which is 3.7 times higher than for Scenario II, while the inference time was with 932 µs even 6.4 times higher. In contrast, for the baseline model the times required for one Gibbs sampling iteration and for inference were much lower than for Scenarios I and II with about 80s and 62ms, respectively. However, the average training time for the baseline model was with 133 minutes for all 10,000 situations still larger due to the high number of situations. The results confirm that the proposed framework can be used in real-time human-agent interactions, while this is not the case for the baseline model, although for the first time the inference time is in the realm of milliseconds and would therefore theoretically enable the baseline model to do inference during human-agent interactions. However, the timing analyses for Scenarios I, II, and III show that the main factor that contributes to slow sampling and inference times for the baseline model are high dimensional CRs, which will be unavoidable when using a grounding framework in an embodied agent, so that the baseline model will not be able to do inference in real-time and will therefore not be useful for embodied agents deployed in human-centered environments. In contrast, the main factor for slower situation processing and inference times for the proposed framework are the number of modalities and the complexity of sentences. For the latter the sentences used in Scenario III are a proper representation for sentences in human-centered environments, while the number of modalities would be higher but based on the current timings, even if they double they would not prevent the usability of the framework in real world interactions.



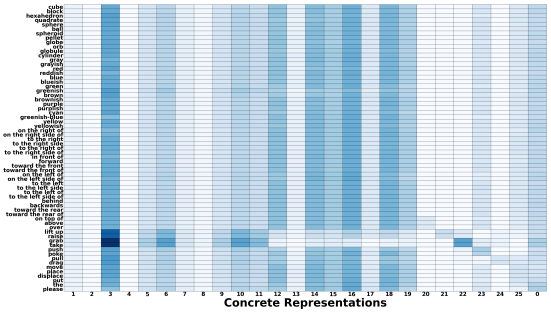
(a) Confusion matrix for the proposed model and TTS100.



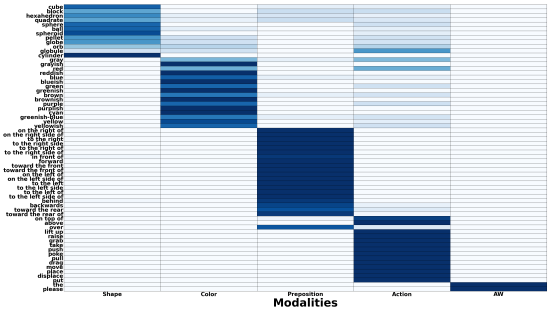
(b) Confusion matrix for the proposed model and TTS100.



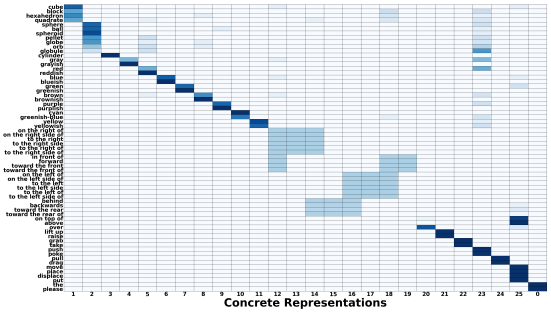
(c) Confusion matrix for the baseline model and TTS100.



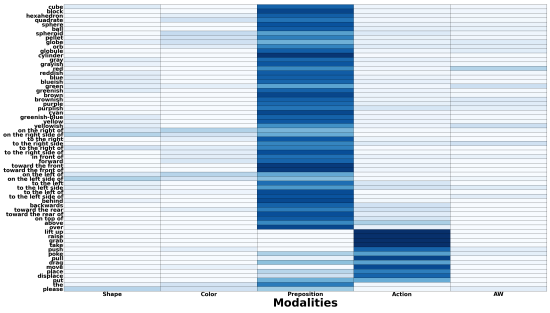
(d) Confusion matrix for the baseline model and TTS100.



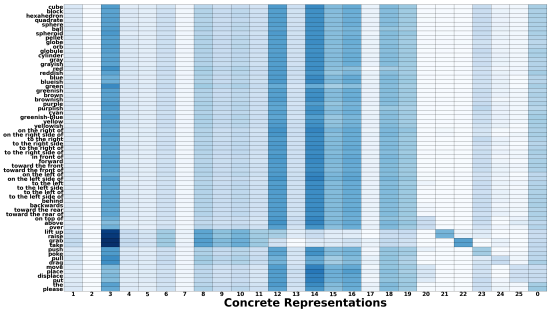
(e) Confusion matrix for the proposed model and TTS60.



(f) Confusion matrix for the proposed model and TTS60.



(g) Confusion matrix for the baseline model and TTS60.



(h) Confusion matrix for the baseline model and TTS60.

Figure 3.17: Confusion matrices showing how often each word of Scenario III was grounded through which modality and CR.

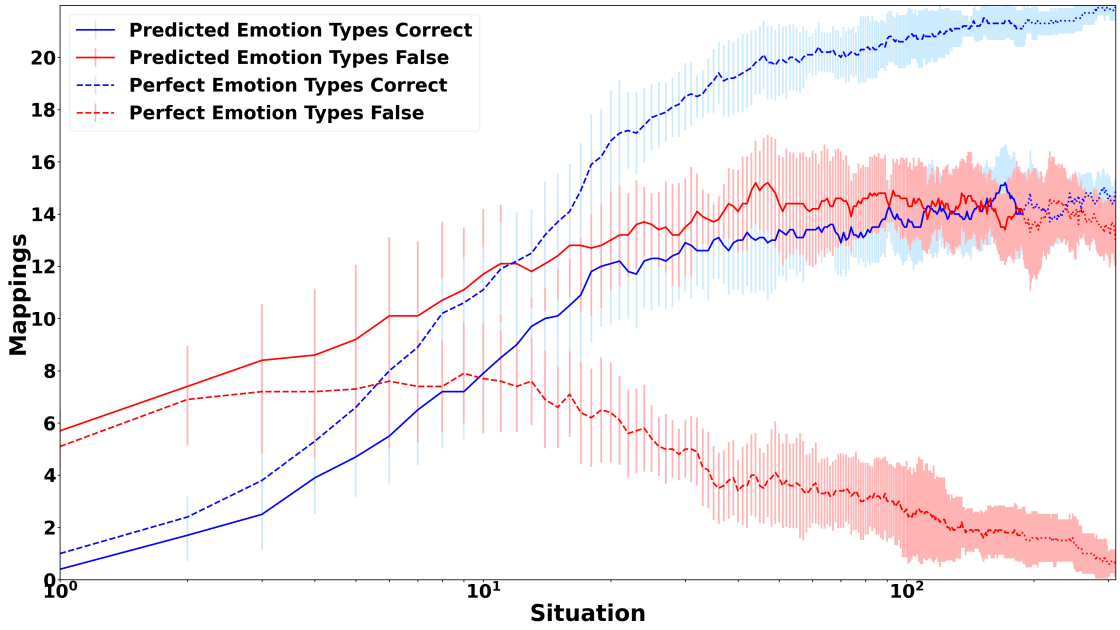


Figure 3.18: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 312 situations of Scenario IV. The continues line represents the results when the predicted CRs are used for all modalities (PRET), while the dashed line represents the results when perfect CRs are used for emotion types (PERT) to investigate the influence of the CR accuracy on the grounding performance of the proposed model. For all lines, the dotted parts only occur when all situations are used for training (TTS100).

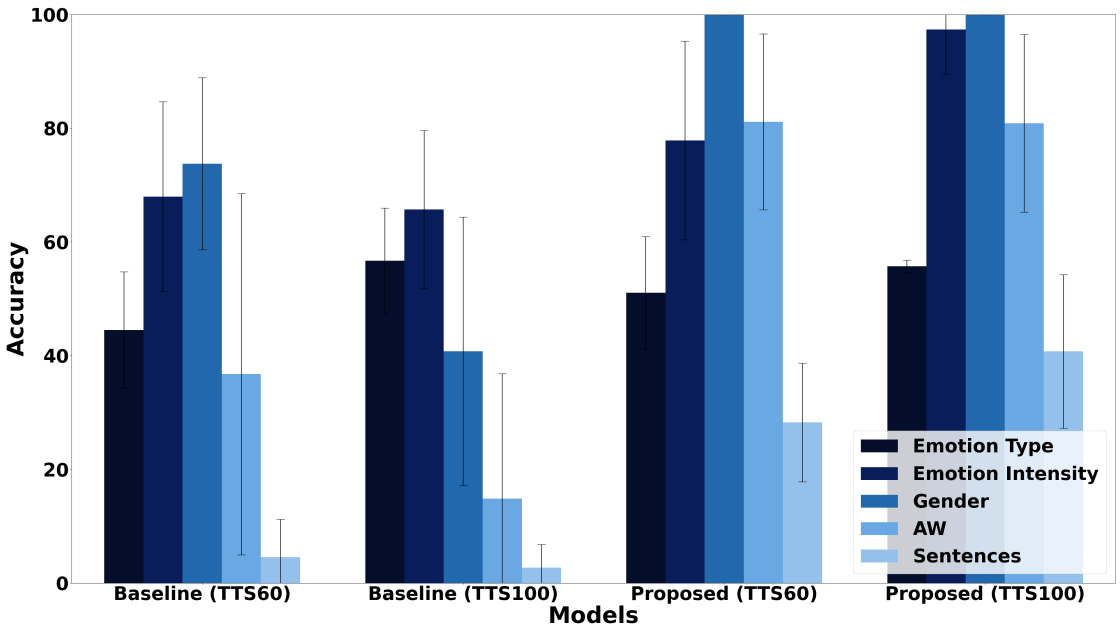
Overall, the results for the third scenario show that the difference in grounding accuracy is not due to the different clustering algorithms used to create CRs but due to the different grounding algorithms used by the proposed and baseline framework. Additionally, the results also show that the proposed framework can handle many different modalities in parallel and a variety of sentences structures including long sentences with two preposition and three color and shape words.

3.6.4 Scenario IV: RAVDESS

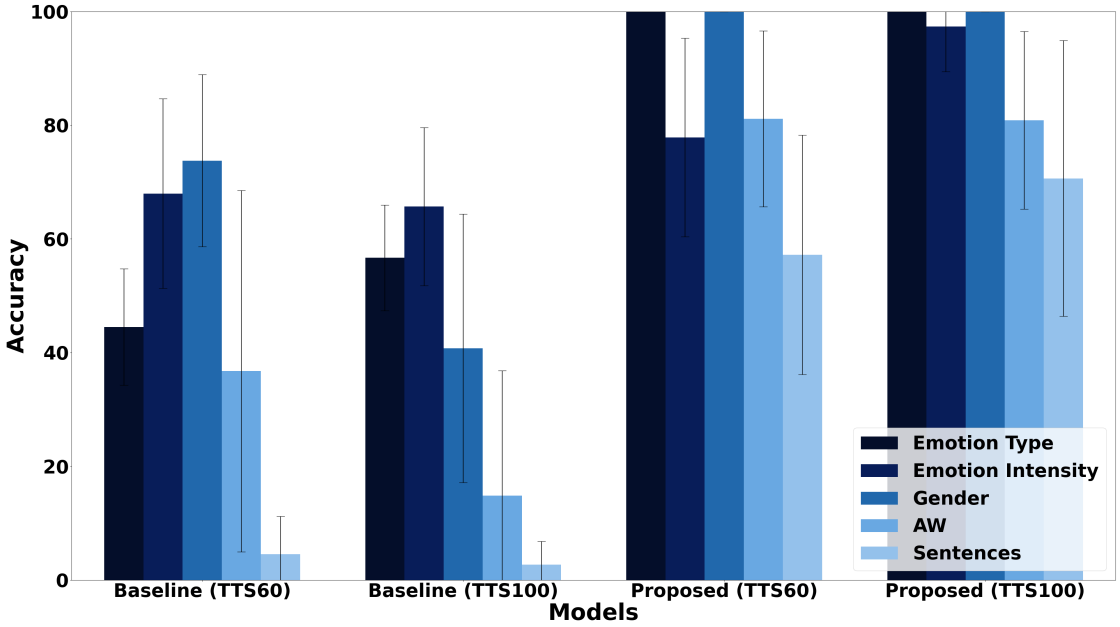
For all previous scenarios the proposed framework employed clustering algorithms, more specifically DBSCAN, to obtain CRs for all percepts in the encountered situations. However, other learning algorithms can also be used. For the fourth scenario the perceptual information is too difficult for clustering algorithms so that deep learning was used instead, as described in Section (3.5.4.1). This is also the main motivation for this scenario, together with the goal to illustrate that grounding is not just relevant for object manipulation tasks but also for many other tasks, such as social interactions which

require understanding of emotions and genders.
Figure (3.18) shows how the mean number of correct and false mappings obtained by the proposed framework changes over all 312 situations. It shows two different cases, which differ regarding the CRs used for emotion types, i.e. for the first case (PRET), the predicted CRs are used, while for the second case (PERT), perfect CRs are used to investigate the effect of the accuracy of the CRs on the grounding performance.
For PRET, represented by continuous lines, the number of correct mappings quickly increases from zero to about twelve mappings for the first 20 situations, and continuous to increase more slowly afterwards to 15 mappings, while the number of false mappings starts with about six mappings and increases over the course of 45 situations to 15 mappings, after which it slowly decreases to 13 mappings. The main reason for the large number of false mappings is that the CRs used for emotion types are highly inaccurate, with an accuracy of 59.6%, while, at the same time, 60% of the employed words refer to them. This assumption is confirmed when looking at PERT, represented by the dashed line, which shows the number of correct and false mappings when perfect CRs are used for emotion types, while the predicted ones are still used for the other two modalities, i.e. emotion intensity and gender.
For PERT, the proposed framework obtains 17 and 20 correct mappings within the first 20 and 45 situations, respectively. If the framework is only allowed to learn during 60% of the situations, it obtains 21 correct mappings, while it obtains one more mapping, i.e. 22, if it continues learning for the remaining situations. In contrast, the number of false mappings increases slightly from five to seven from the first to the second situation, stays stable for about eight situations and decreases then continuously to two mappings after 60% of the situations have been encountered and one mapping after all situations have been encountered. Both cases together illustrate that the proposed grounding algorithm depends on the accuracy of the obtained CRs, however, it does not require perfectly accurate representations because it is able to obtain all correct mappings for the second case, although the CRs for emotion intensities and genders only have accuracies of 73.5% and 89.8%, respectively.

Figure (3.19) shows the accuracies for the proposed and baseline models, all modalities, both test splits, and PRET as well as PERT. It shows that the proposed model achieves a higher accuracy than the baseline model in all cases, i.e. for all modalities, train/test splits and both CRs of emotion types, except for emotion types, when the predicted CRs are used and all situations are encountered during training. In fact, for genders, the proposed model achieves perfect grounding due to the high accuracy of the corresponding CRs, i.e. 89.8%. The figure also confirms the results in Figure (3.18) that the grounding accuracy improves with the number of encountered situations, which seems intuitive but is not necessarily the case, as shown by the results obtained for the baseline model, i.e. the latter obtained less accurate groundings for most modalities



(a) Results when the predicted CRs are used for all modalities.



(b) Results when perfect CRs are used for emotion types.

Figure 3.19: Mean grounding accuracy results and corresponding standard deviations for both grounding models, train/test splits, and all modalities of Scenario IV. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

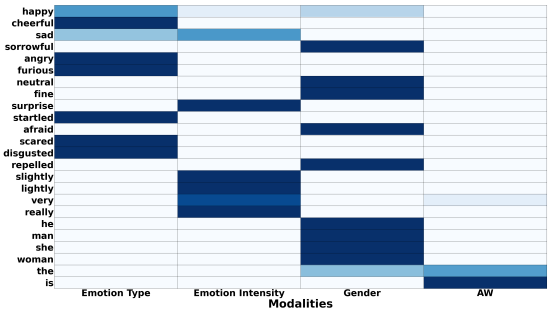
when using all situations for training and testing due to the larger number of situations in the test set. For the baseline model, using perfect CRs for emotion types increases the accuracy of the groundings obtained for emotion types and genders as well as the ac-

curacy of AWs, although the accuracy of the latter two only increases for TTS100, while the accuracy of the emotion intensity groundings decreases independent of the number of situations encountered during training.

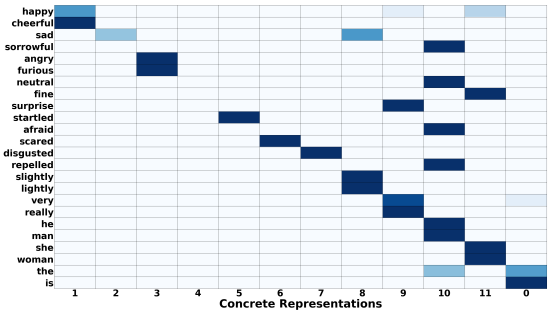
Although the accuracies provide a good overview of how accurate the groundings for each modality are, they do not provide any details about the wrong groundings or the accuracy of the groundings obtained for individual words. Therefore, the left side of Figure (3.20) shows the confusion matrices for all words and modalities, which illustrate how often each word was grounded through the different modalities and highlight two interesting points. First, both models show a high confusion for emotion types, i.e. all of them have non-zero probabilities to be mapped to CRs representing emotion intensities or genders, due to the low accuracy of the corresponding CRs for TTS60. The confusion decreases for TTS100, in which case most words converge to one modality for the proposed model, i.e. only "happy" and "sad" are still confused as a gender or emotion intensity, respectively. However, this does not lead to a substantial increase in grounding accuracy for emotion types because some words, e.g., "surprise" and "afraid", converge to the wrong modality so that the probability to be mapped to a CR of an emotion type decreases to zero.

The right side of Figure (3.20) shows confusion matrices of words and different CRs, thereby allowing to investigate whether the CR a word is grounded through is correct, which might not be the case if there is a high confusion between CRs of the same modality. The fourth column, representing the emotion type neutral, is very noticeable in Figures (3.20b, 3.20h, and 3.20d) because both models do not map any word to it, except for the proposed model and TTS60 (Figure 3.20b). However, even in the latter case, the probability that the word "fine" gets mapped to it is very low because most of the time it is mapped to the CR of the concept FEMALE (column 11). Otherwise, the results show that, for the proposed model, the confusion is normally across modalities and not between CRs of the same modality. In contrast, the baseline model shows strong confusions between CRs of the same modality, e.g., for TTS60 "happy" and "disgusted" are more often grounded through anger than happiness and disgust, respectively.

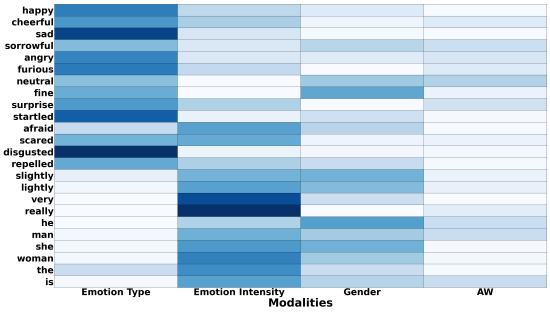
The average time it took the proposed framework to process a new situation and update its mappings was 4.58ms, while the inference time was only 64.82 µs. In contrast, one Gibbs sampling iteration of the baseline model took 655ms. Since 100 iterations were used, the average training time for the baseline model was 65s for all 312 situations, while the inference time was on average 7.64ms for each situation. These results confirm that the timings of the proposed framework are mostly influenced by the complexity of the encountered utterances and the number of modalities, therefore, the timings are very similar to the timings for Scenario I. In contrast, for the the baseline model the timings mostly depend on the dimensionality of the employed CRs leading to timings similar to Scenario III, which also used one-hot encoded vectors for CRs.



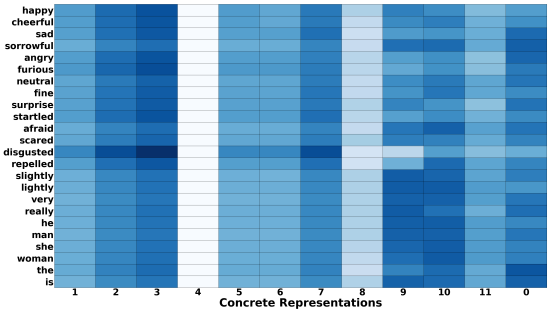
(a) Confusion matrix for the proposed model and TTS100.



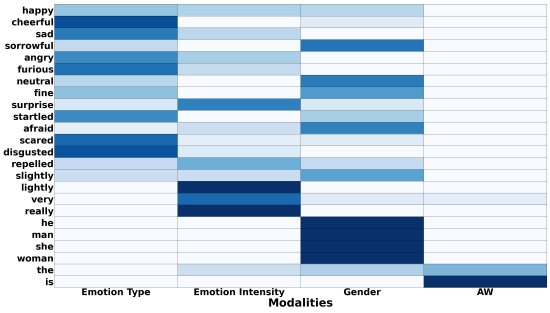
(b) Confusion matrix for the proposed model and TTS100.



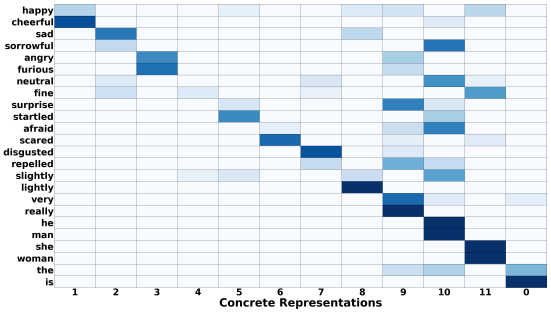
(c) Confusion matrix for the baseline model and testing TTS100.



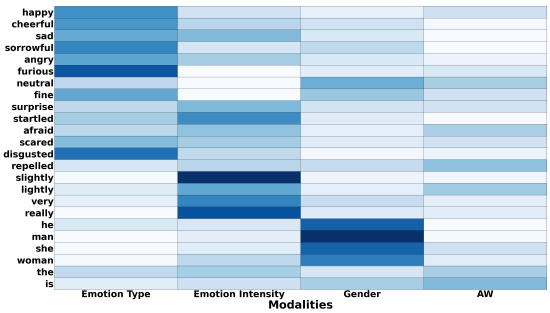
(d) Confusion matrix for the baseline model and TTS100.



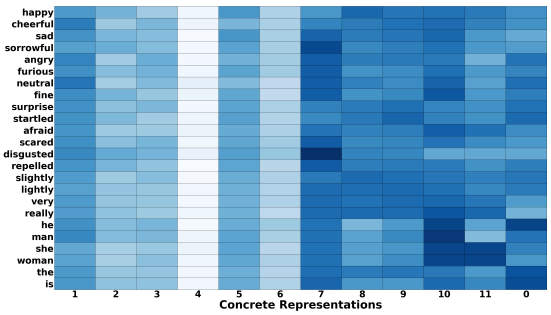
(e) Confusion matrix for the proposed model and TTS60.



(f) Confusion matrix for the proposed model and TTS60.



(g) Confusion matrix for the baseline model and TTS60.



(h) Confusion matrix for the baseline model and TTS60.

Figure 3.20: Confusion matrices showing how often each word of Scenario IV was grounded through which modality and CR.

Overall, the evaluation shows that the proposed model is able to ground higher level concepts, like emotion types or genders, and that it can also employ non-clustering algorithms, in this case deep neural networks, to extract CRs. Otherwise, the results confirm the results obtained for the previous scenarios, i.e. the proposed framework outperforms the baseline in terms of AW detection and grounding accuracy as well as its abilities to learn continuously without requiring explicit training. The latter does not only make it more applicable for real-world scenarios but also more transparent, because it is possible to observe how a new situation influences the obtained groundings.

3.7 Discussion

Due to the dynamicity and unpredictability of human-centered environments as well as the ambiguity of natural language, language grounding frameworks must be able to ground language in a continuous and open-ended manner, need to be able to cope with synonymy and homonymy, and should not rely on external support. Therefore, in this chapter, a novel CSL based unsupervised grounding framework was proposed (Section 3.3). The proposed framework was evaluated through four different scenarios (Section 3.5) that differ based on the used modalities, words, CRs, and number of situations. The obtained results showed that the framework can be used to detect AWs and ground non-AWs and phrases through corresponding CRs in an unsupervised manner. Furthermore, the obtained results also showed that the proposed framework is able to outperform a state-of-the-art Bayesian learning model based on the achieved grounding accuracy, while at the same time being able to obtain new groundings continuously and in an open-ended manner. Due to Scenario III it is clear that the difference in grounding accuracy is not due to the different mechanisms employed to obtain CRs. Scenario IV also illustrates that the proposed framework is able to employ any kind of algorithm to obtain CRs and is therefore more flexible than the probabilistic model, which always requires the use of KMeans clustering, even if a different form of CRs had already been obtained, e.g. for Scenario IV the class labels obtained by the employed deep neural networks.

Finally, the results also showed that the proposed framework is able to handle the ambiguity of language in form of synonyms and homonyms. Additionally, the timing analysis showed that the proposed framework is able to process new situations quick enough for real-world deployment, while the baseline model requires much more time because it would have to re-train every time from scratch.

Overall, the results presented in this chapter show that the proposed framework can be used to ground words in an unsupervised manner in a variety of scenarios differing in the complexity of the encountered language as well as percepts. The main concern is the scalability when considering that even Scenario III has only 63 words, which is

relatively small in comparison to the number of words used by humans in normal conversations, but the unsupervised framework was already not able to ground all of the words. This is especially concerning since Scenario III consisted of 10,000 situations, which is a large number of situations in comparison to the relatively small number of employed words and CRs, so that it can be ruled out that just providing more situations would be helpful. Thus, in the next chapter (Chapter 4) the unsupervised grounding framework will be extended with a mechanism to learn from feedback provided by an external agent, to increase its sample-efficiency and the accuracy of the obtained groundings.

4 Enhancing Unsupervised Grounding through Optional Feedback

4.1 Motivation

Unsupervised CSL based grounding frameworks, like the one proposed and evaluated in the previous chapter (Chapter 3), do not require any support from another agent to successfully ground words through corresponding CRs. However, for more complex and realistic scenarios, like Scenarios II and III in Chapter (3), the previously proposed unsupervised framework is not able to ground all words successfully. Thus, the question arises whether the utilization of some form of supervision, such as feedback, could improve the accuracy and sample-efficiency of the grounding framework (see Section 3.7 for a detailed discussion), which leads to the following research questions:

1. How to combine unsupervised and supervised grounding mechanisms to enable an artificial autonomous agent to utilize the support provided by other agents without depending on it?

2. Does extending the unsupervised grounding framework proposed in the previous chapter (Chapter 3) with a mechanism to handle external support increase the sample-efficiency of the framework and the accuracy of the obtained groundings?

3. Which type of feedback, i.e. only non-verbal feedback or verbal and non-verbal feedback, has the most positive effect, i.e. leads to the highest sample-efficiency and accuracy?

4. How to handle wrong feedback[1] so that artificial autonomous agents still learn the correct groundings, even if all provided support is wrong?

The rest of this chapter answers above questions by proposing a framework that combines CSL and IL by extending the unsupervised grounding framework described in the previous chapter (Chapter 3) with a mechanism to handle two different types of support, i.e. non-verbal and verbal feedback. Section (4.2) provides an overview of

[1] Wrong feedback can be accidental, due to malicious intent of the supporting agent, or due to noisy or corrupted input, e.g. the learning agent might misunderstand to which object the supporting agent is pointing.

previously proposed supervised learning models as well as previous attempts to combine unsupervised and supervised grounding approaches. Afterwards, Section (4.3) describes the proposed grounding framework. The two scenarios used to evaluate the proposed framework, the employed evaluation criteria as well as the obtained results are described in Sections (4.4 and 4.5). Finally, Section (4.6) concludes this chapter with a final discussion of the research questions, a summary of the main contributions, and an outlook towards possible future work to address observed limitations.

4.2 Related Work

The motivation for supervised grounding approaches comes from the fact that, although infants and young children do not need any support to learn their native language, there is evidence that active support by their parents or other language proficient people simplifies word learning and therefore makes children learn faster [11, 39, 8]. Inspired by these studies, supervised or interactive grounding approaches try to utilize the support of a tutor to obtain word-CR mappings in a sample-efficient and highly accurate manner. The main idea is that direct teaching and feedback prevents an artificial agent from learning wrong mappings and reduces the complexity of language grounding by limiting the number of possible mappings.

For example, She et al. [88] and She and Chai [87] investigated the use of a dialog system to ground higher-level actions, like “pick up”, “grab” or “stack”, through already grounded lower-level actions or manipulation sequences modifying the gripper of the robot employed in their study, like “open”, “move”, or “close”. However, not only the words referring to the modifications of the gripper state or location were assumed to be already grounded through their corresponding actuator commands, also the colors and shapes that the human tutor used to refer to the manipulation objects were assumed to be already grounded. While the proposed framework was able to achieve perfect grounding when the higher-level actions were taught step by step through the already grounded lower-level actions, it only worked due to the strong assumption that the groundings of both the lower-level actions and object characteristics already existed, which cannot be assumed when deploying artificial agents in human-centered environments. Additionally, the study also made the implicit assumption that the human tutor knows what the robot knows, i.e. which words have already been grounded and only uses these words to teach the unknown words. Neverthless, if additional mechanisms are available to ground the lower-level actions and object characteristics in an unsupervised manner, allowing teaching of higher-level actions through a situated dialog can help learning them faster and without mistakes.

Misra et al. [55] followed a similar approach, i.e. grounding higher-level actions through already grounded lower-level actions. The employed system was able to successfully

ground actions like “distribute”, “mix”, or “arrange” through lower-level actions but relied again on the availability of grounded lower-level actions and objects, thus, being only useful in combination with a mechanism that is able to ground lower-level actions and object characteristics.

Cakmak et al. [13] investigated the benefit of active supervised learning over passive supervised learning. In the context of grounding, active supervised learning means that the learner can ask the tutor about the label for a specific object, while passive supervised learning means that the tutor decides when and in which order new groundings are taught. In their study, 24 participants taught four concepts to a robot through natural language using a predefined grammar so that the robot was able to discard all words except the word referring to the target object. Interestingly, the participants could also show an object and ask for its name or provide negative examples by showing a different object and stating a word that does not refer to it, e.g. showing a house and saying “snowman”. The grounding mechanism did not require any groundings to work, however, indirectly only a single word was provided because all auxiliary words were already known and automatically discarded so that it is not clear whether it would work with more realistic sentences.

Lopes and Chauhan [49] followed an interactive learning approach using an incremental one-class learning algorithm. In the conducted study, the interaction was controlled by a human tutor who could either teach the name of an object to a robot or ask the robot about the name of an object and if the name was incorrect, provide the correct name so that the robot could update its mappings. The study only used a single modality representing the shape of the objects as perceptual input and single words as linguistic input, therefore, it is not clear whether the algorithms would work for more realistic scenarios with natural language utterances consisting of multiple words and more complex perceptual input.

Bleys et al. [10] and Spranger [94] employed the Grounded Naming Game methodology to ground single words referring to the color of objects and spatial relations, respectively. In the employed experiments, two robots were interacting with each other in an environment with two to four objects. One of the robots acted as the tutor knowing the correct groundings, while the other robot acted as the learner trying to learn the correct groundings from the tutor. To teach a color or spatial relation, the tutor said the name of the color of one of the objects or described one of the objects through its spatial relation. In response, the learner utilized previously learned groundings to determine which object the tutor referred to and pointed to it or if it had not learned the corresponding groundings, it randomly pointed to one of the objects. Afterwards, the tutor signaled success or failure depending on whether the learner pointed to the correct object. In the latter case, the tutor pointed to the correct object so that the learner knew the correct grounding at the end of the interaction. Grounding success was evaluated based on the

number of successful interactions. In both studies the learner relatively quickly learned the correct groundings, leading to a high number of successfull interactions, and illustrating the efficiency of the Grounded Naming Game methodology. The main drawback of the used methodology is the dependency on a supporting agent who already knows the correct groundings, and is able and willing to support the learning agent as well as on the correctness of the provided support, which both cannot be guaranteed.

One possibility to overcome this limitation would be to combine unsupervised and supervised grounding approaches, however, so far this has not received much attention despite the potential to combine their strengths and eliminate or at least reduce the impact of their shortcomings. Nevens and Spranger [58] investigated the combination of cross-situational and interactive learning and came to the conclusion that the more feedback is provided, the faster new mappings are obtained and the higher the accuracy of the obtained mappings. While these findings, i.e. that feedback improves the accuracy and sample-efficiency, seem reasonable and intuitive, the employed cross-situational learning algorithm was very limited, thus, it is not clear whether feedback would have provided the same benefit, if a more sophisticated unsupervised grounding mechanism would have been employed.

A different study by Roesler [70] extended an unsupervised CSL based grounding framework, which has achieved state-of-the-art grounding performance [72], with a mechanism to learn from explicit teaching and showed that explicit teaching increases the convergence speed towards the correct groundings. The main disadvantage of the employed supervised learning mechanism is that it requires the tutor to artificially create a special teaching situation, which is a simplified version of the environment specifically designed to ensure that the agent will correctly learn a specific mapping. Since finding a tutor who is able and willing to put this amount of effort into teaching the agent is very unlikely, the approach is not really applicable for real human-agent interactions.

Due to the fact that in both studies one of the employed mechanisms, i.e. the unsupervised mechanism in [58] and the supervised mechanism in [70], were quiet limited, the framework presented in this chapter combines two mechanisms that have previously been shown to achieve state-of-the-art grounding results individually and evaluates whether their combination leads to better sample-efficiency and accuracy, while ensuring at the same time that supervision can be provided in a simple and natural way, and is not required to learn the correct groundings. Additionally, the impact of incorrect feedback on the grounding performance of the proposed framework is also investigated because it cannot be assumed that the provided support is always correct, e.g. due to noisy input or malicious intent of the supporting agent. For example, a study conducted by Nomura et al. [62] showed that children might harm or abuse a robot out of curiosity and not because they want to cause serious harm. Thus, it is very likely that some people will try to trick the learning agent through incorrect support, when

deployed in human-centered environments without any professional supervision.
The proposed framework and the results for the case when only correct feedback is provided for Scenario II have already been published in [73]. The proposed framework has also been extended with a mechanism to learn from explicit teaching in [70], however, due to the limitations of explicit teaching the conducted study is not included in this chapter.

4.3 A Feedback Enhanced Unsupervised Grounding Framework

This section describes a novel grounding framework that combines unsupervised and supervised grounding components. More specifically, it describes several extensions to the framework proposed in Chapter (3) to enable it to learn from non-verbal and verbal feedback by a human tutor to improve its sample-efficiency and grounding accuracy, while simultaneously ensuring that the model does not require feedback, i.e. it is still able to ground words through corresponding CRs when no feedback is provided, and still works when wrong feedback is provided.
The proposed framework consists of three main parts: (1) CR creation component (Section 4.3.1), which converts percepts to CRs utilizing a standard clustering algorithm, (2) Unsupervised grounding component (Section 4.3.2), which detects AWs and word-CR mappings through CSL, (3) Supervised grounding components (Section 4.3.3), which utilize two different interactive feedback based learning mechanisms to improve the accuracy of word-CR mappings as well as the acquisition speed. The unsupervised grounding component is based on the unsupervised grounding framework proposed in the previous chapter (Section 3.3) that has been shown to outperform a state-of-the art probabilistic model based grounding approach (Sections 3.4 and 3.6). The individual parts of the proposed novel hybrid grounding framework are illustrated below and in Figure (4.1), while they are described in detail in the following subsections.

1. **Concrete representation creation component**:
   - **Input**: Percepts.
   - **Output**: CRs of percepts.
2. **Cross-situational learning component**:
   - **Input**: Natural language instructions, CRs, previously detected AWs, and word and CR occurrence information.
   - **Output**: Set of AWs and word to CR mappings.
3. **Interactive learning component**:
   - **Input**: Natural language instructions, CRs, AWs, and feedback information.
   - **Output**: Word to CR mappings.

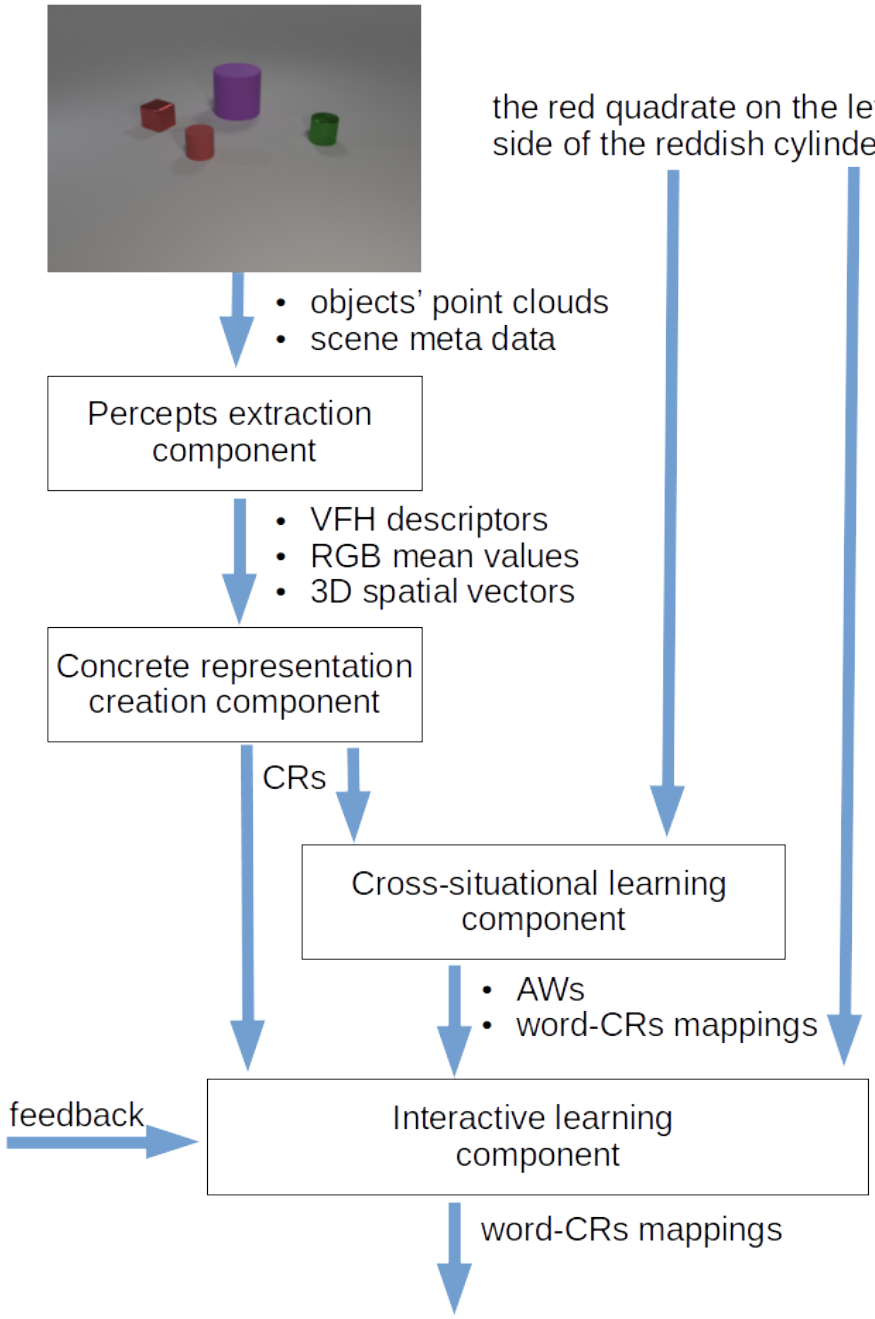


Figure 4.1: Illustration of the components of the proposed framework and the data flow for the second scenario (Section 3.5.2). First percepts, i.e. VFH descriptors, RGB mean values, and 3D spatial vectors, are extracted using the point clouds of the objects in the current scene and the meta-data generated by the scene extraction script (see Section 3.5.2 for details). Afterwards, corresponding CRs are obtained, which are then provided as input to the CSL and IL components. Both components also take as input the natural language sentence, while the IL component also receives as input the AWs and mappings obtained by the CSL components as well as any feedback information available, which can be both verbal or non-verbal feedback (see Section 4.3.3 for details). Finally, the IL component outputs the word-CRs mappings based on both co-occurrence information and available feedback.

4.3.1 Concrete representation creation

Since the proposed framework is an extension of the unsupervised grounding framework proposed in Chapter (3) it uses the same CR creation component, which allows great flexibility because it does not require the use of a specific clustering or classification algorithm and leads to explicit CRs that enhance the explainability and transparency of the grounding framework. A detailed explanation of the CR creation component is provided in Section (3.3.1) of the previous chapter.

4.3.2 Cross-situational learning

The CSL component of the proposed framework consists of the same mechanisms as the unsupervised grounding framework proposed in the previous chapter, thus, for a detailed description of the employed unsupervised mechanisms, i.e. the unsupervised AW detection and grounding mechanisms, please refer to the corresponding sections in the previous chapter, i.e. Sections (3.3.2 and 3.3.3), respectively.

4.3.3 Interactive learning

The supervised or IL component is inspired by the "Naming Game" methodology [97], but has been designed so that it smoothly integrates with the unsupervised grounding component described in the previous section (Section 4.3.2). The main idea is to allow agents to receive and utilize non-verbal and verbal feedback from a tutor, when available, to speed up the grounding process and improve the accuracy of the obtained groundings. The integration with the unsupervised CSL based grounding mechanism is crucial to avoid that the agent requires feedback to learn new mappings and also to improve its robustness to incorrect feedback. The supplied feedback can consist of two parts: (1) pointing to the correct object, which allows the agent to identify the percepts belonging to the target object, and (2) an utterance, which provides a short description of the characteristics of the target object. While the first part, i.e. pointing to the correct object, is required for the feedback mechanism to work, the second part, i.e. the utterance, is optional. The feedback is used by the agent to update its mappings to increase the probability that it identifies the target object correctly in similar situations in the future.

Algorithm (6) provides an illustration of the two proposed feedback mechanisms. First, the set of non-target CRs (*NOCR*) is calculated by subtracting the set of target object CRs (*TOCR*) from the set of all object CRs (*AOCR*). Afterwards, word-CR and CR-word feedback pairs are created or updated for each word in the instruction sentence and each CR in *TOCR*, if no verbal feedback is available. The reason for this is that due to the available non-verbal pointing-based feedback, it is clear which object the instruc-

Algorithm 6 The feedback procedure takes as input the words of the instruction of the current situation (WI) and the feedback sentence (WF), the set of all object CRs ($AOCR$), the set of the target object CRs ($TOCR$), the set of detected AWs (AWS), and the sets of previously obtained word-CR feedback ($WCRPSF$) and CR-word feedback ($CRWPSF$), and returns updated $WCRPSF$ and $CRWPSF$.

procedure FEEDBACK($WI, WF, AOCR, TOCR, AW, WCRPSF, CRWPSF$)
  $FRC = 2$
  $AOCR \setminus TOCR \rightarrow NOCR$
  if WF is $\varnothing$ **then**
    for w in $(WI - AW)$ **do**
      for p in $TOCR$ **do**
        $WCRPSF_{w,cr} += FRC$
        $CRWPSF_{cr,w} += FRC$
  else
    for w in $(WF - AW)$ **do**
      for p in $TOCR$ **do**
        $WCRPSF_{w,cr} += FRC$
        $CRWPSF_{cr,w} += FRC$
      for p in $NOCR$ **do**
        $WCRPSF_{w,cr} -= FRC$
        $CRWPSF_{cr,w} -= FRC$
  return $WCRPSF, CRWPSF$

tion refers to and therefore which CRs the instruction words refer to. Otherwise, i.e. if verbal feedback is also provided, feedback pairs are created or updated using the feedback sentence and each CR in $TOCR$ and $NOCR$. More specifically, mappings from the words in the feedback sentence to the CRs of the target object are strengthened, while the mappings from the feedback words to all other CRs are weakened. Thus, the feedback mechanism automatically takes into account verbal feedback (WF), if available, but does not require it because otherwise the instruction words (WI) will be used instead. When verbal-feedback is provided, the feedback mechanism does not only increase the values of the mappings from the words in WF to the CRs in $TOCR$ but also decreases the values of the mappings in WF to the CRs in $NOCR$, thereby, following the assumption that most of the times the other objects have not the same color and shape of the target

Algorithm 7 High-level overview of the unsupervised grounding procedure described by Algorithm (4) to highlight the integration of the feedback mappings (lines 5 and 8).

1: **procedure** GROUNDING($W, CR, WCRPS, CRWPS, AWS, PP, WO, CRO, PMS$)
2: Substitute words with phrases from PP
3: Update AWS (Algorithms 1, 2, and 3) and remove AW from W
4: Update $WCRPS$, and $CRWPS$ using W and CR
5: $WCRPS \cup WCRPSF$
6: **for** w in $WCRPS$ **do**
7: Save highest $WCRP$ to GW
8: $CRWPS \cup CRWPSF$
9: **for** $j = 1$ to CR_number **do**
10: Save highest $CRWP$ to GCR
11: **return** $GW \cup GCR$

object.

The feedback mechanism has one parameter, i.e. FRC, which represents the feedback related change and determines how strong the influence of feedback is on the obtained mappings. FRC was initially set to 2 to ensure that feedback is twice as important as co-occurrence information, while ensuring that wrong feedback would not have a too strong influence. This setting was later also experimentally verified as the best setting. Feedback is integrated with the unsupervised algorithm by merging $WCRPS$ and $WCRPSF$ as well as $CRWPS$ and $CRWPSF$ in lines 5 and 9 of Algorithm (7) so that pairs that received positive feedback are strengthened and pairs that received negative feedback are weakened.

4.4 Experiments

The proposed framework (Section 4.3) is evaluated through modified versions of the second and third scenarios presented in Section (3.5) because the other two scenarios only contain one target object or person so that no target selection and therefore feedback by a tutor is possible. The employed utterances and percepts are exactly the same, while the interaction procedure is slightly different due to the availability of support from another agent. For all scenarios two different types of support are investigated, i.e. non-verbal feedback and combined verbal and non-verbal feedback. Additionally, different feedback rates as well as different amounts of wrong feedback are investigated. The remainder of this section will explain the extended experiment procedure, while an overview of the employed words and percepts can be found in Section (3.5) since only the experimental procedure is different.

In all scenarios the experimental procedure is as follows:

1. A scene is generated and the agent determines the geometric characteristics and

colors of all objects as well as their spatial relationships.

2. A description of one of the objects is provided to the agent mentioning its shape, color and spatial relation to another object at the beginning or end of the situation, which is also described by its shape and color. For Scenario II no action is executed, thus, the reference is always describing the position of the target object at the beginning of the situation. In contrast, for the third scenario, the reference can describe either the initial position of the manipulation object or the target position. For Scenario III, it is also possible that two references, i.e. the first for the initial position and the second for the target position, are provided.

3. The agent updates its groundings and utilizes them to select (Scenario II) or manipulate (Scenario III) the target object.

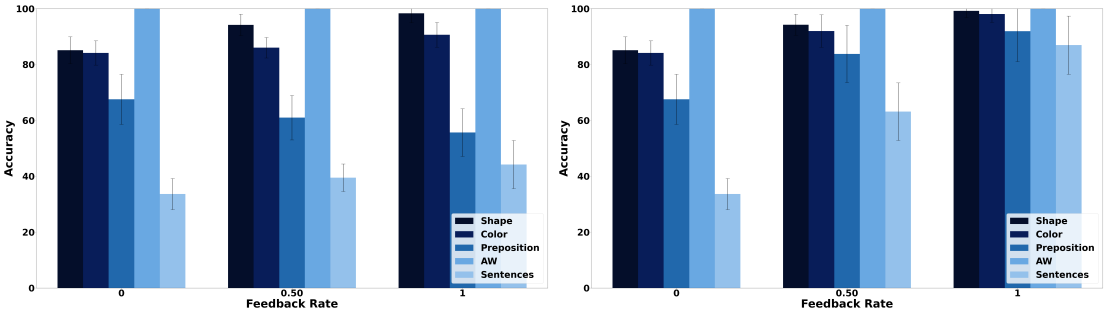
4. (optional) The tutor provides non-verbal feedback by pointing to the correct object or a combination of non-verbal and verbal feedback, e.g. "yes the red cylinder" or "no the red cylinder".

5. (optional) The agent updates its groundings based on the received feedback.

Steps 4 and 5 are optional because feedback is not always provided by the tutor, in which case the supervised learning mechanism of the proposed framework will have no effect. Two different validity cases are investigated for both scenarios and both types of feedback. In the first case, the tutor is always providing correct feedback, while in the second case the feedback is sometimes incorrect. There can be different reasons for incorrect feedback including a misinterpretation by the agent of where the tutor is pointing or the tutor is pointing to a wrong object, e.g. by accident, out of curiosity how the agent will cope with wrong feedback, or due to malicious intent. For this experiment, incorrect pointing-only feedback means that the tutor points to one of the objects that are not mentioned in the utterance, while for combined pointing and verbal feedback the tutor points to one of the non-target objects while mentioning color and shape words that do not refer to the actual color and shape of that object.

4.5 Results

In the following subsections the groundings obtained by both the proposed framework (Section 4.3) and the baseline framework (Section 3.3) for the two investigated scenarios (Section 4.4) are presented and evaluated. The baseline framework is the unsupervised grounding framework proposed in the previous chapter because it has shown to perform better than other state-of-the-art frameworks for the investigated scenarios. Since



(a) Grounding results when correct pointing-only feedback is provided.

(b) Grounding results when both correct pointing based and verbal feedback is provided.

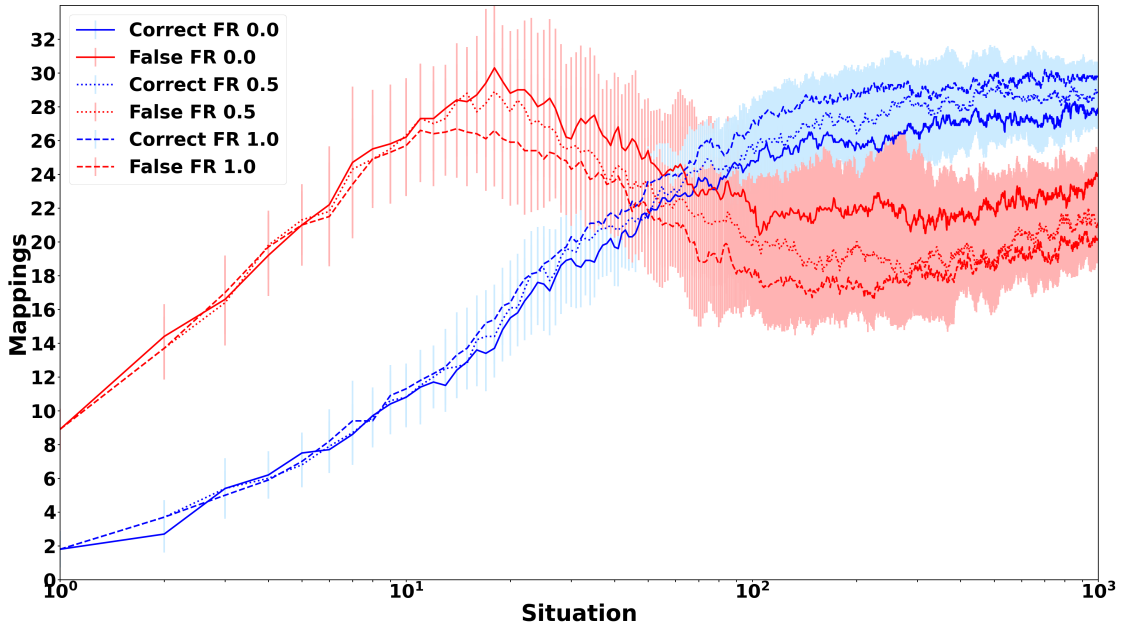
Figure 4.2: Mean grounding accuracy results, corresponding standard deviations, and percentage of sentences for which all words were correctly grounded for both types of feedback of Scenario II, when all feedback is correct.

the same utterances and percepts are provided in the same sequence to both frameworks, any difference in grounding performance can only be due to the use of feedback by the proposed framework. Both frameworks receive situations one after the other as if processing the data in real-time during the interaction because they do not require explicit training phases, therefore, all situations are used for training and testing.
The main questions investigated are (1) Whether the proposed feedback mechanisms improve the sample efficiency of the framework and the accuracy of the obtained groundings, (2) Whether verbal feedback is important, i.e. leading to better sample efficiency and grounding accuracy than only non-verbal feedback, and (3) Whether the proposed feedback mechanisms lead to a decrease in grounding accuracy, if wrong feedback is provided. All three questions are investigated for both scenarios.

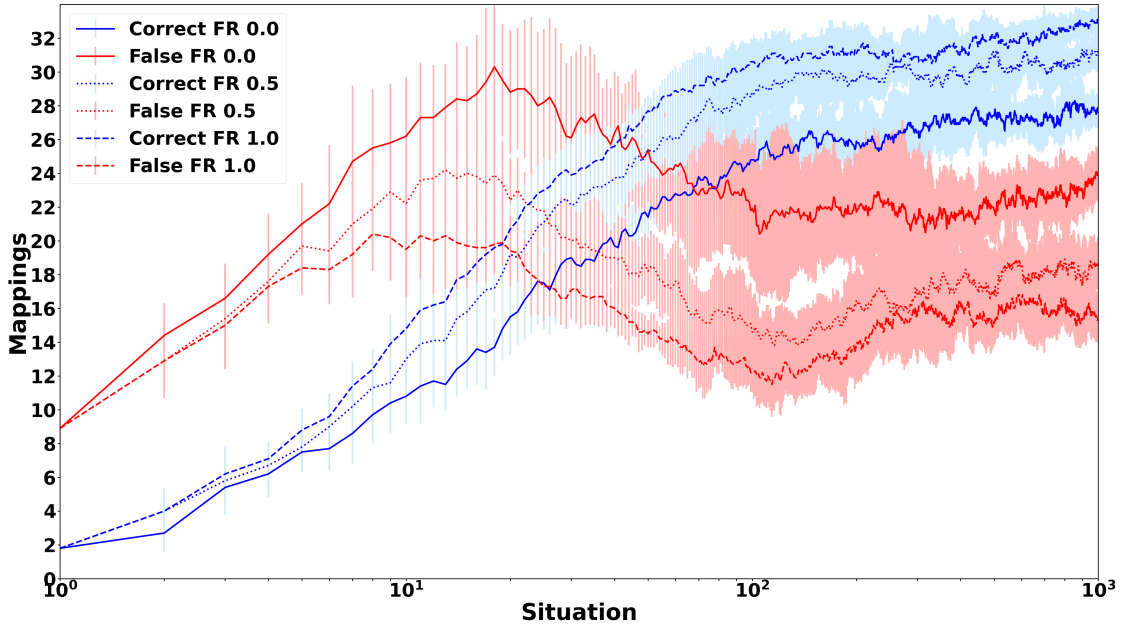
4.5.1 Scenario II: CLEVR - Correct Feedback

This section presents the results obtained for Scenario II (Section 3.5.2) for both feedback types when all provided feedback is correct, thereby, allowing the investigation of the first two questions for Scenario II.
Figure (4.2a) shows the grounding results when only non-verbal pointing-only feedback is provided. It shows that pointing-only feedback has only a small mostly positive effect on the accuracies of shape and color groundings as well as a light negative effect on preposition groundings so that number of sentences for which all words were correctly grounded increases by about 10%. In comparison, when the tutor also provides verbal feedback, the accuracy of the obtained groundings improves visibly for all modalities (Figure 4.2b) so that the number of correctly grounded sentences increases from about 30% to more than 85%. This increase is mostly due to an increase in the grounding accuracy of prepositions by more than 20%, while the accuracy of shape and

color groundings increases only by about 10%, which could also be due to the fact that the shape and color groundings were already more accurate, when no feedback was provided.
The results show that both feedback types have a positive effect on the overall grounding accuracy for Scenario II and that combining non-verbal and verbal feedback is essential for the accuracy of preposition groundings, while it has only a limited effect on the accuracy of shape and color groundings. The reason is that pointing-only feedback only directly influences shape and color groundings but has also indirectly a negative effect on preposition groundings, while verbal feedback has a direct positive effect on the accuracy of preposition groundings.
Figure (4.3) shows how the number of correct and false mappings changes over all 1,000 situations for different feedback rates, i.e. depending on how often feedback is given. For pointing-only feedback the final number of correct mappings, i.e. after 1,000 situations, increases by on average one and two mappings, when feedback is provided for 50% or 100% of the situations. Similarly, the number of false mappings decreases by on average one and four mappings, when feedback is provided for 50% or 100% of the situations (Figure 4.3a). For combined pointing and verbal feedback there is a clear difference regarding the final grounding accuracy as well as the speed correct mappings are obtained (Figure 4.3b). For example, when no feedback is provided, it takes nearly 80 situations until the number of correct mappings is equal to the number of false mappings, while it only takes about 25 and 19 situations when feedback is given for 50% of the situations or all situations, respectively. Additionally, after all 1,000 situations have been encountered the number of correct mappings is about 10% higher if feedback is provided on average every second situation than if no feedback is provided, followed by another 6% increase, if feedback is provided every situation.
These results illustrate the benefit of verbal feedback in addition to pointing feedback and the benefit of supervised grounding in addition to unsupervised grounding in terms of both grounding accuracy and sample-efficiency. However, the results also show that the framework does not depend on feedback and achieves decent grounding results, if no feedback is provided, which is important because the availability of feedback cannot be guaranteed. Figure (4.3) also illustrates the online learning ability of the proposed framework, which is very important when considering deployment in real environments that require open-ended learning because it is impossible to create a large enough dataset that contains all possible words and CRs that an agent could encounter. In addition, it also shows the transparency and explainability of the framework because at any time it is possible to check the current mappings and understand why



(a) Pointing-only feedback.



(b) Pointing and verbal feedback.

Figure 4.3: Mean number and standard deviation of correct and false mappings over all 1,000 situations of Scenario II, when correct feedback is provided for 0%, 50% or 100% of the situations, where FR means feedback rate.

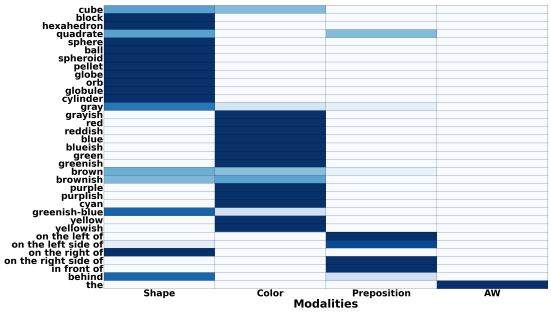
they have been created based on the available co-occurrence information stored in WCRPS, CRWPS, WCRPSF, and CRWPSF (Sections 4.3.2 and 4.3.3).

While the accuracies and the numbers of correct and false mappings presented in Figures (4.2 and 4.3) provide a good overview of how accurately the groundings are for

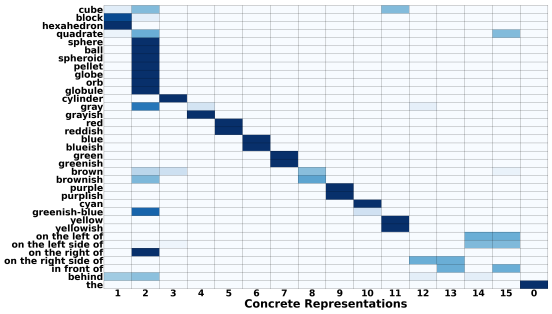
each modality and how the number of correct and incorrect mappings changes over time, they neither provide any details about the accuracy of the groundings obtained for individual words nor any details about the wrong mappings. Therefore, Figure (4.4) shows the confusion matrices for all words and modalities as well as all words and CRs illustrating how often each word was grounded through the different modalities and CRs, respectively. Figure (4.4a), which shows the confusion matrix for all words and modalities, when no feedback is provided, illustrates that there is some inter-modality confusion between shapes and colors as well as prepositions and shapes, while overall most words are grounded through the correct modality. When pointing-only feedback is provided for every situation the inter-modality confusion for shapes and colors decreases (Figure 4.4c), while for prepositions it increases slightly, which is consistent with the accuracy results in Figure (4.2a). Most inter-modality confusion disappears when combined verbal and pointing feedback is provided every situation (Figure 4.4e), i.e. there is only very light inter-modality confusion for "cube", "brownish", "in front of", and "behind".

Since grounding is not about determining the modality a word belongs to but to create a mapping from words to corresponding CRs, it is important to also look at the confusion matrices of words over different CRs. Figure (4.4b) shows the confusion matrix of words over different CRs when no feedback is provided. The figure shows that there is not much intra-modality confusion and that most of the inter-modality confusion is for CR 2 because many words are incorrectly mapped to it, although all mappings except for "on the right of", "greenish-blue", and "gray" are relatively weak. For the preposition words it is interesting to see that most of them are mapped to two CRs, which is correct because all prepositions should be grounded through two homonymous CRs.

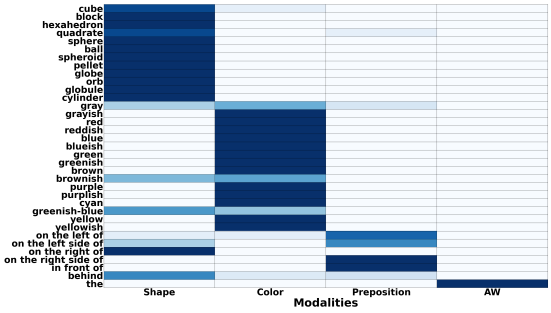
When looking at Figure (4.4c), which shows the confusion matrix for the case were pointing-only feedback is provided for every situation, it is interesting to see that the mappings for prepositions are less accurate and weaker. The reason for this is that pointing-only feedback strengthens the mappings from the CRs of the target object's shape and color with all words of the utterance. Thus, the mappings from the preposition words to the CRs of shape and color are strengthened as well, the former even more because there are only three different CRs for shapes in comparison to eight for colors. However, when combined verbal and pointing feedback is provided, the confusion for prepositions is nearly completely gone and in general there is no intra-modality and only slight inter-modality confusion (Figure 4.4f). The large improvement for prepositions is due to the availability of the feedback sentence which ensures that only the mappings from the color and shape words of the target object to the corresponding CRs are strengthened. This clearly shows the importance of verbal feedback when comparing it to the pointing-only feedback case, while the confusion matrices also showed that the proposed framework is also able to achieve decent groundings, if no feedback is



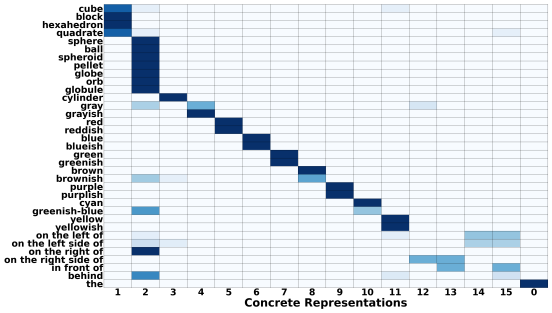
(a) Confusion matrix of words over different modalities when no feedback is provided.



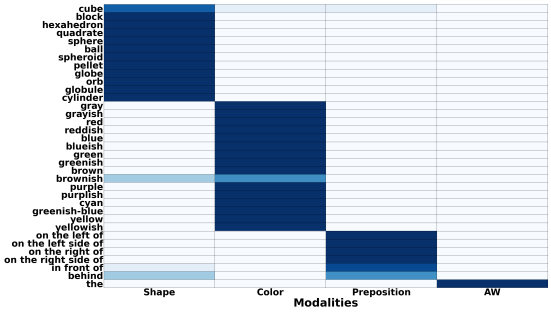
(b) Confusion matrix of words over different CRs when no feedback is provided.



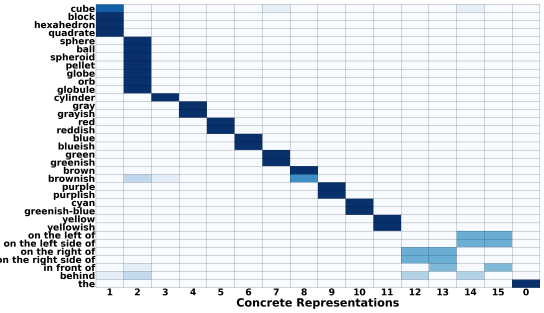
(c) Confusion matrix of words over different modalities when pointing-only feedback is provided for every situation.



(d) Confusion matrix of words over different CRs when pointing-only feedback is provided for every situation.



(e) Confusion matrix of words over different modalities when combined pointing and verbal feedback is provided for every situation.



(f) Confusion matrix of words over different CRs when combined pointing and verbal feedback is provided for every situation.

Figure 4.4: Confusion matrices for all ten situation sequences and three different types of interactions, i.e. no feedback, pointing-only feedback and combined verbal and pointing feedback, of Scenario II, when all feedback is correct.

available, which confirms the results presented in Section (3.6).

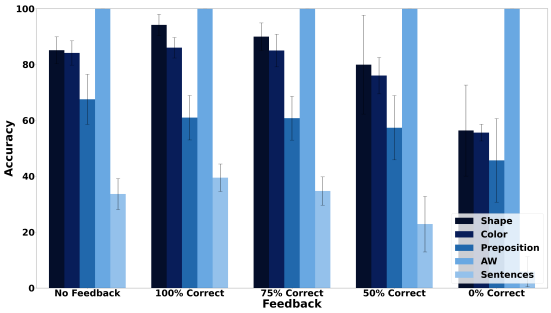
Overall, the results for Scenario II show that both feedback mechanisms improve the sample-efficiency of the original framework and lead to more accurate groundings, while the integration with the unsupervised grounding mechanism ensures that the framework is still able to achieve decent grounding results, when no feedback is available. Furthermore, the results also show that combined verbal and pointing feedback

provides a substantial benefit over pointing-only feedback in terms of grounding accuracy and sample-efficiency. Especially, for modalities that are not benefiting by pointing-only feedback, i.e. prepositions in case of Scenario II, verbal feedback is important to prevent the creation of incorrect mappings. The remaining question, which will be investigated in the next section (Section 4.5.2), is how robust the framework is in regard to incorrect feedback, i.e. how does incorrect feedback affect the grounding accuracy and sample-efficiency of the framework and how much does the effect of incorrect feedback depend on the number of situations for which feedback is provided and the percentage of incorrect feedback.

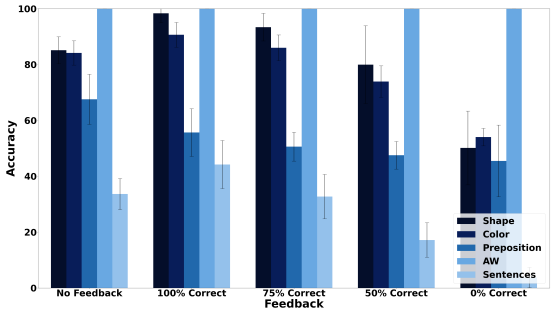
4.5.2 Scenario II: CLEVR - Incorrect Feedback

This section presents the results obtained for Scenario II (Section 3.5.2) for both feedback types when part of the provided feedback is incorrect, thereby, allowing the investigation of the third question for Scenario II.
Figures (4.5a and 4.5b) show that the grounding accuracy decreases slightly when incorrect feedback is provided for 50% of the situations in comparison to no feedback, independent of how often feedback is provided, i.e. every situation or every second situation. When all feedback is incorrect the accuracy decreases by more than 30% for all modalities. For combined verbal and pointing feedback the results are overall similar, however, interestingly the overall grounding accuracy still improves slightly when feedback is provided every situation but only 50% of it is correct. This illustrates that the combined verbal and pointing feedback is more robust in regard to incorrect feedback because only when more than 50% incorrect feedback is provided every second situation or always incorrect feedback every situation the accuracy is worse than if no feedback is provided at all. Thus, while these results highlight the negative effect incorrect feedback can have, they show at the same time that even if incorrect feedback is provided every situation, the framework is still able to learn decent groundings for most of the modalities. Important to note is also that even 25% incorrect feedback is rather unrealistic when deploying a robot over a long time in human environments and for this amount of incorrect feedback both types of feedback either increase the accuracy of groundings (combined verbal and pointing feedback) or reach the same accuracy as if no feedback would be provided (pointing-only feedback).

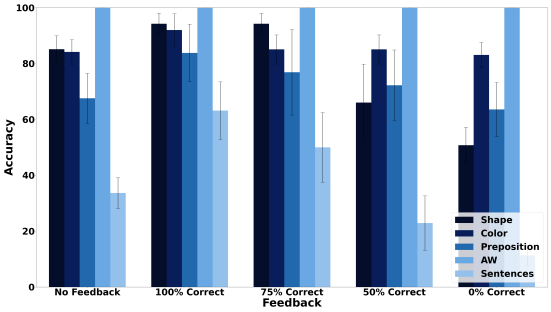
These findings are supported by Figure (4.6a) which shows that the number of correct mappings still increases slightly when 25% of pointing-only feedback is incorrect, while for 50% or more incorrect feedback the number of correct mappings after encountering all 1,000 situations of Scenario II drops by up to 25% in comparison to the case when no feedback is provided. For combined pointing and verbal feedback, the number of correct mappings is higher than when no feedback is provided, even if 50% of the feed-



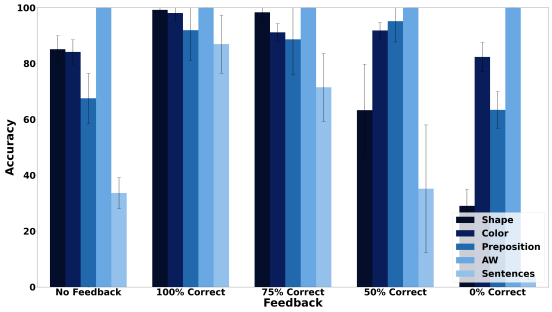
(a) Grounding results when pointing-only feedback is provided for 50% of the situations.



(b) Grounding results when pointing-only feedback is provided for all situations.



(c) Grounding results when combined pointing and verbal feedback is provided for 50% of the situations.

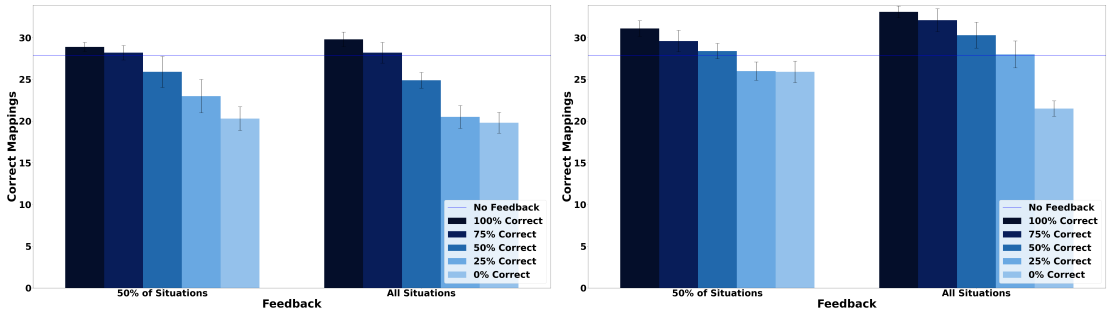


(d) Grounding results when combined pointing based and verbal feedback is provided for all situations.

Figure 4.5: Mean grounding accuracy results, corresponding standard deviations, and percentage of sentences for which all words were correctly grounded for Scenario II for both types of feedback and different rates of correct and wrong feedback.

back is incorrect and if feedback is provided every situation, even if 75% of the feedback is incorrect. Thus, combined verbal and pointing feedback only has a slightly negative effect, i.e. the number of correct mappings decreases by about 7% (2 mappings). When feedback is provided every situation, incorrect combined verbal and pointing feedback only has a negative effect if nearly all feedback, i.e. at least more than 75%, is incorrect, which leads to a 20% decrease of correct mappings in comparison to the case without feedback. These results clearly show the robustness of the grounding framework due to the use of both unsupervised and supervised grounding mechanisms because the majority of feedback when deploying an agent in real world will be correct, which means that feedback will have either a positive or in the worst case no effect on the number of correct mappings.

Figure (4.7) illustrates which words, modalities, and CRs are mostly affected by incorrect feedback. Figures (4.7c and 4.7d) show that the highest increase in inter-modality confusion is for colors and prepositions, while for shapes the confusion increases for



(a) Pointing-only feedback.

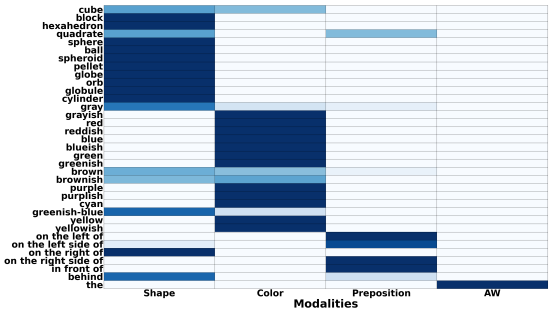
(b) Pointing and verbal feedback.

Figure 4.6: Mean number and standard deviation of correct mappings after encountering all 1,000 situations of Scenario II for different percentages of correct feedback, when feedback is provided for 50% or all of the situations, and for both feedback types.

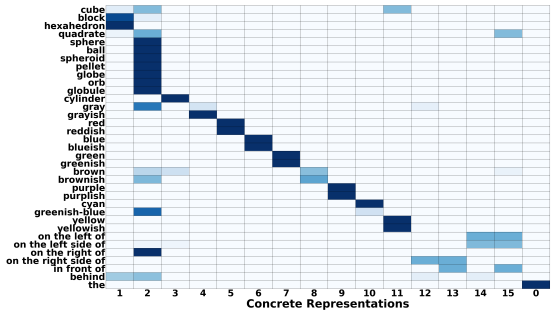
some words, e.g. "cylinder", while it decreases at the same time for other words, e.g. "cube". Intra-modality confusion only increases for the shape words referring to the concept of a *Cube*. The higher level of inter-modality confusion for color and preposition words is due to them being grounded through CRs of shapes, which can be explained by the fact that all words of the provided utterance are mapped to the CRs of the shape and color of one of the non-target objects. The reason for the increase of intra-modality confusion for shapes can be explained by the fact that there are much less shape CRs than color CRs so that the words are still mapped to the shape CRs but often to the wrong CR because the tutor is always pointing to one of the non-target objects.

When incorrect combined verbal and pointing feedback is provided every situation the confusion for colors and prepositions does not increase (Figure 4.7e), which confirms the accuracy results shown in Figure (4.5d), while the confusion for shape words increases strongly. Figure (4.7e) shows that most confusion across modalities is with the CR of the concept CYLINDER including strong intra-modality for shapes, i.e. most words referring to the concept SPHERE are at least partially mapped to the CR of CYLINDER. Additionally, there is strong inter-modality confusion for shapes because many of the shape words are partially grounded through preposition CRs and the word "cylinder" is wrongly detected as an AW for both feedback types.

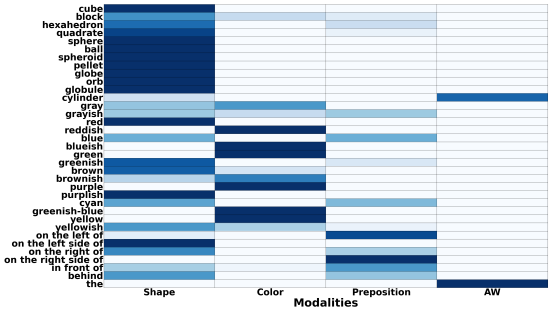
The results show that the robustness of the framework in regard to incorrect feedback depends on how much of the provided feedback is actually incorrect. For example, if 25% of the feedback is incorrect it negates the positive effect of pointing-only feedback, while for combined verbal and pointing feedback even 50% incorrect feedback provides still a benefit in terms of the number of correct mappings (Figure 4.6b), therefore, illustrating that adding verbal feedback to pointing feedback does not only increase the benefit of the provided feedback but also makes it more robust regarding incorrect



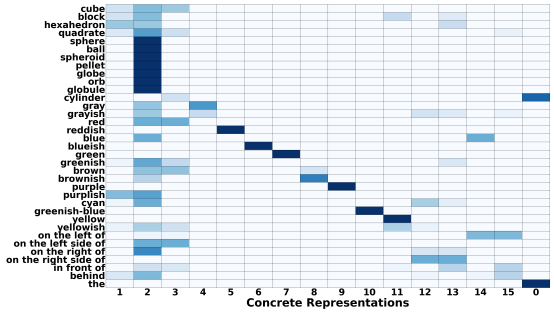
(a) Confusion matrix of words over different modalities when no feedback is provided.



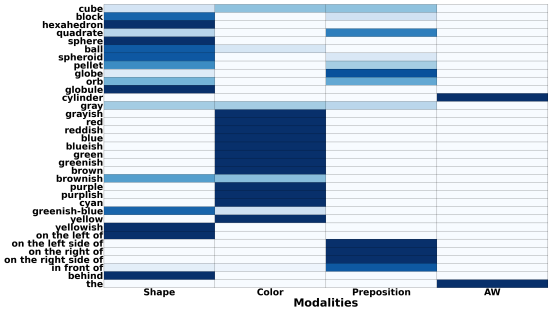
(b) Confusion matrix of words over different CRs when no feedback is provided.



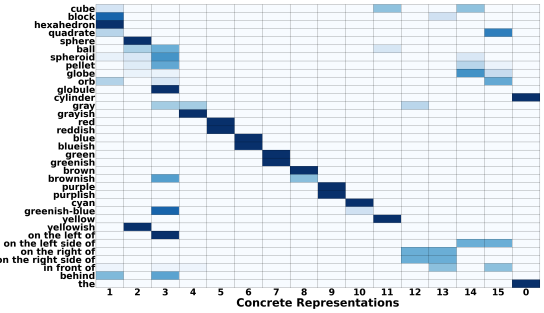
(c) Confusion matrix of words over different modalities when pointing-only feedback is provided for every situation.



(d) Confusion matrix of words over different CRs when pointing-only feedback is provided for every situation.



(e) Confusion matrix of words over different modalities when combined pointing and verbal feedback is provided for every situation.

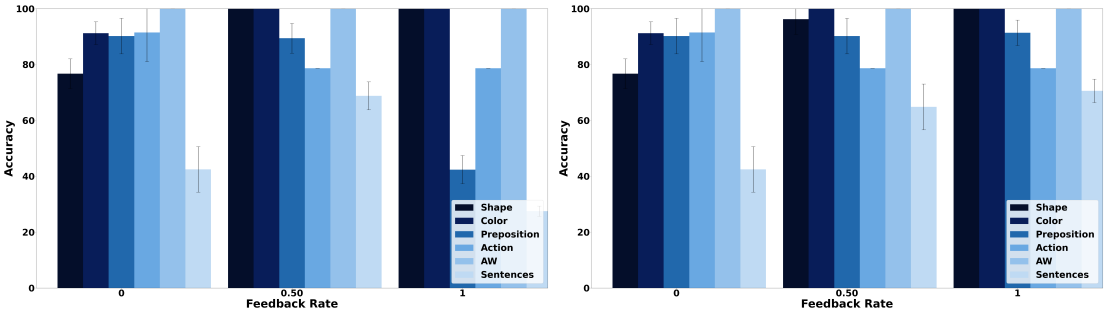


(f) Confusion matrix of words over different CRs when combined pointing and verbal feedback is provided for every situation.

Figure 4.7: Confusion matrices for all ten situation sequences and three different types of interactions, i.e. no feedback, pointing-only feedback and combined verbal and pointing feedback, of Scenario II, when all feedback is incorrect.

feedback.

The obtained results illustrate that the benefit of feedback outweighs the potential damage caused by incorrect feedback, especially since it is very unlikely that more than 25% of the feedback will be incorrect when an agent would interact with many different people in a variety of situations, i.e. a few people might provide incorrect feedback by accident or to trick the agent but they will represent much less than 25% of the people



(a) Grounding results when correct pointing-only feedback is provided.

(b) Grounding results when both correct pointing based and verbal feedback is provided.

Figure 4.8: Mean grounding accuracy results, corresponding standard deviations, and percentage of sentences for which all words were correctly grounded for both types of feedback of Scenario III, when all feedback is correct.

the agent would interact with.

4.5.3 Scenario III: Synthetic - Correct Feedback

This section presents the results obtained for Scenario III (Section 3.5.3) for both feedback types when all provided feedback is correct, thereby, allowing the investigation of the first two questions for Scenario III, which includes more modalities, words, CRs, and situations than Scenario II, while the used CRs are perfect due to the use of synthetic percepts, i.e. one-hot encoded vectors (Section 3.5.3 provides a detailed description of the scenario).

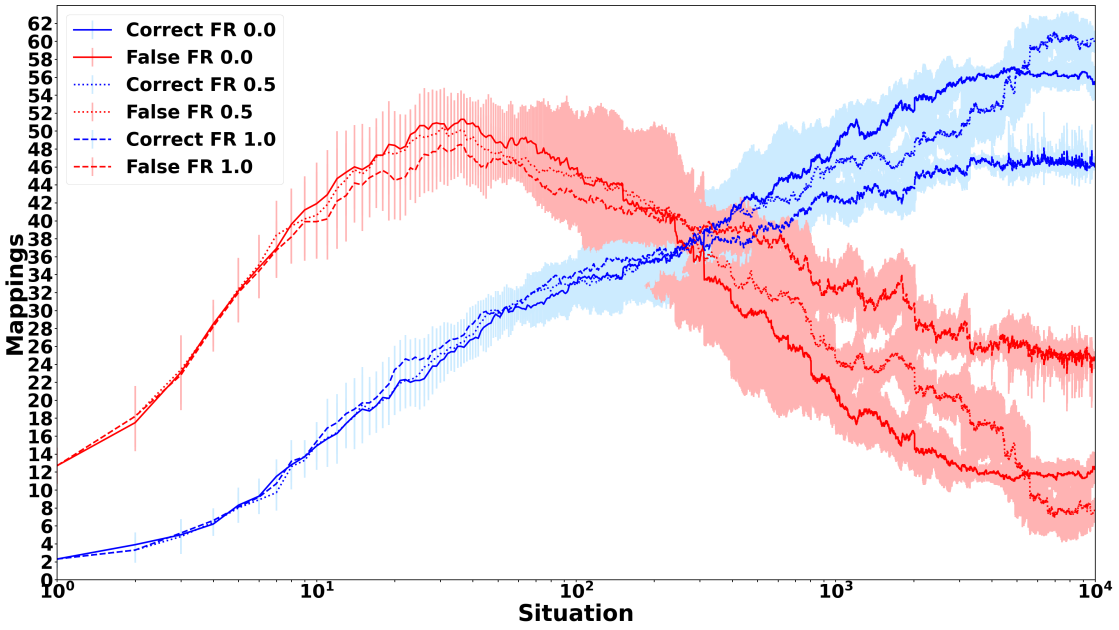
Figure (4.8a) shows the grounding results when only non-verbal pointing-only feedback is provided. It shows that how often feedback is provided has a strong impact on how the accuracy of groundings is affected because pointing-only feedback increases the accuracy of shape and color groundings, while it has a negative impact on the accuracy of preposition and action groundings. The reason is, as already explained in Section (4.5.1), that pointing-only feedback strengthens the mappings from the CRs of the target object's shape and color with all words of the utterance so that the mappings from the preposition and action words to the CRs of the target object's shape and color are also strengthened. Due to the fact that all shape and color words are correctly grounded when pointing-only feedback is only provided for 50% of the situations, increasing the number of situations for which feedback is provided only has a negative effect because it decreases the accuracy for preposition and action groundings. Interestingly, the percentage of sentences for which all words were correctly grounded is lower when feedback is provided for all situations than if no feedback is provided, while providing feedback only for every second situation leads to a large increase in accuracy from about 42% to more than 70%.

For combined verbal and non-verbal feedback (Figure 4.8b) the accuracy of shape and color groundings improves when feedback is provided for 50% of the situations, the accuracy of actions decreases slightly, and the accuracy of prepositions does not change. Increasing the feedback rate to 100% only has a light positive influence on the accuracies of the shape and preposition groundings, while the accuracies for colors and actions stay the same. This shows that combined verbal and pointing feedback makes the influence of the feedback more robust than pointing-only feedback and has also, in contrast to pointing-only feedback, no negative influence on the accuracy of preposition groundings.

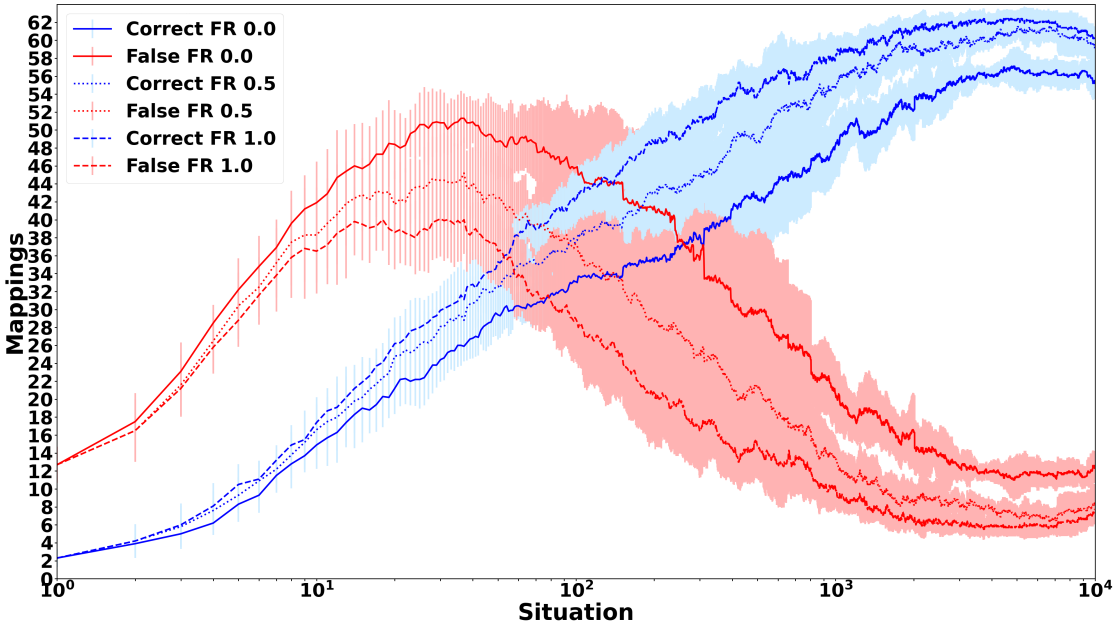
Figure (4.9) shows how the number of correct and false mappings changes over all 10,000 situations for different feedback rates, i.e. depending on how often feedback is given. For pointing-only feedback the final number of correct mappings, i.e. after 10,000 situations, increases when feedback is provided for 50% of the situations by about 8% but decreases by more than 16%, if pointing-only feedback is provided for all situations (Figure 4.9a). When also verbal feedback is provided for half the situations the number of correct mappings increases by nearly 7%, while providing combined feedback for the remaining situations increases the number of correct mappings only by another 2% (Figure 4.9b). Additionally, when no feedback is provided it takes nearly 260 situations until the number of correct mappings is larger than the number of incorrect mappings, in comparison to 94 and 36 situations, when combined feedback is provided for 50% and 100% of all situations. Thus, while more feedback has a positive effect on the accuracy of the obtained groundings and how fast they are obtained when combined feedback is provided, too much pointing-only feedback is harmful and can even lead to worse groundings than no feedback. These results illustrate the benefit of verbal feedback in addition to pointing feedback. However, the results also show that the framework does not depend on feedback and achieves decent grounding results, if no feedback is provided, which is important because the availability of feedback cannot be guaranteed. Important to note is also that depending on the type of feedback, more is not always better, which will be interesting when looking at the effect of wrong feedback.

While the accuracies and numbers of correct and false mappings (Figures 4.8 and 4.9) provide a good overview of how accurately the groundings are for each modality, they neither provide any details about the accuracy of the groundings obtained for individual words nor any details about the wrong mappings. Therefore, Figure (4.10a) shows the confusion matrix for all words and modalities, which illustrates how often each word was grounded through the different modalities, when no feedback is provided.

The figure shows that most inter-modality confusion is related to shapes, colors, and prepositions being grounded as actions, while the only inter-modality confusion for actions is for “push” being considered an AW. When pointing-only feedback is provided for every situation the confusion for shapes and colors disappears (Figure 4.10c), while



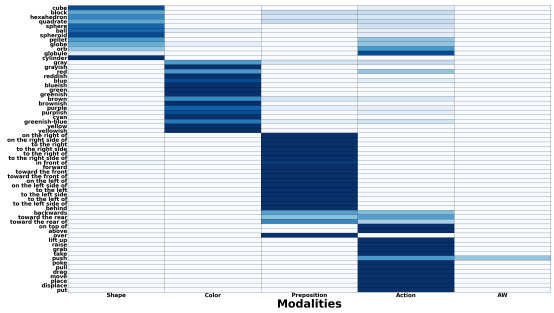
(a) Pointing-only feedback.



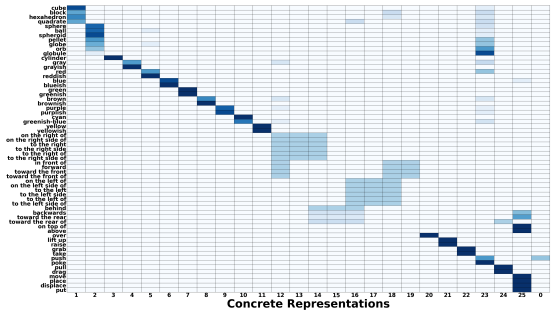
(b) Pointing and verbal feedback.

Figure 4.9: Mean number and standard deviation of correct and false mappings over all 10,000 situations of Scenario III, when correct feedback is provided for 0%, 50% or 100% of the situations, where FR means feedback rate.

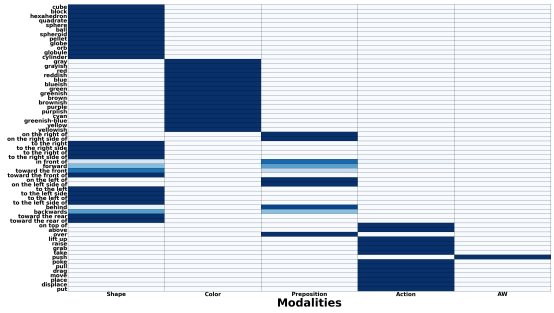
the confusion for prepositions increases strongly so that more than half of the prepositions words are mapped to CRs of shapes and two to CRs of actions. The action word "push" is now also always considered as an AW, while this is only the case for four out of ten runs when no feedback is provided.



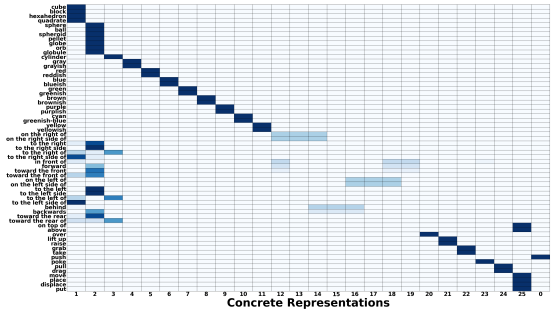
(a) Confusion matrix of words over different modalities when no feedback is provided.



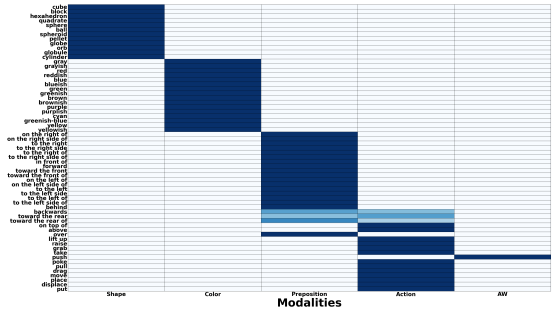
(b) Confusion matrix of words over different CRs when no feedback is provided.



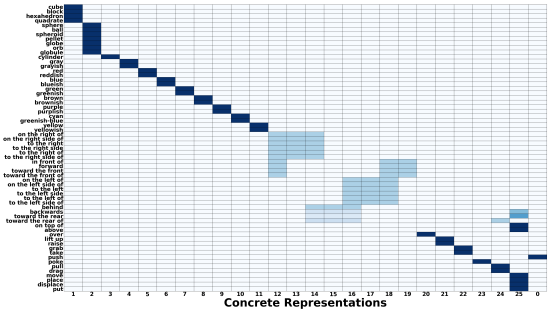
(c) Confusion matrix of words over different modalities when pointing-only feedback is provided for every situation.



(d) Confusion matrix of words over different CRs when pointing-only feedback is provided for every situation.



(e) Confusion matrix of words over different modalities when combined pointing and verbal feedback is provided for every situation.



(f) Confusion matrix of words over different CRs when combined pointing and verbal feedback is provided for every situation.

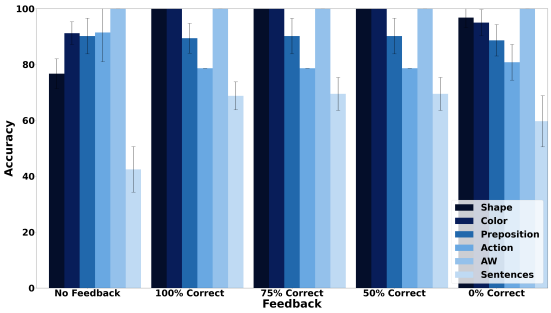
Figure 4.10: Confusion matrices for all ten situation sequences and three different types of interactions, i.e. no feedback, pointing-only feedback and combined verbal and pointing feedback, of Scenario III, when all feedback is correct.

When combined verbal and pointing feedback is provided every situation (Figure 4.10e) there is only light confusion for prepositions and actions, i.e. two preposition words are grounded as actions, one action word is labeled as an AW, and there is light confusion for three preposition words. Figure (4.10b) shows the confusion matrix of words over different CRs when no feedback is provided. The figure shows that there is no intra-modality confusion and that most of the inter-modality confusion is for the CR of the

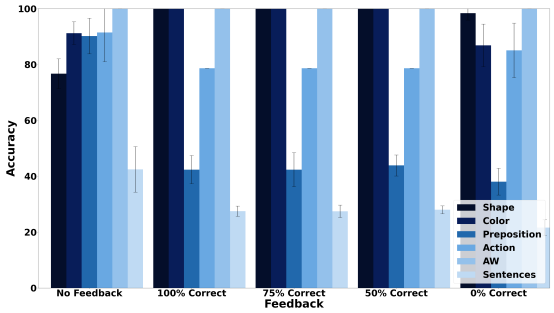
action PUSH, i.e. CR 23, because many shape and color words are incorrectly mapped to it. The confusion matrix for pointing-only feedback (Figure 4.10c) shows also no intra-modality confusion and only inter-modality confusion for prepositions because more than half of the preposition words are ground through shape CRs. The same is the case for combined verbal and pointing feedback, i.e. no intra-modality confusion and in this case also only slight inter-modality confusion due to preposition words being incorrectly grounded through CR 25 of the concept MOVE (Figure 4.10f).
In general, the results obtained for Scenario III confirm the results for Scenario II presented in Section (4.5.1), i.e. both feedback mechanisms improve the grounding accuracy and sample-efficiency of the original unsupervised framework, while the hybrid framework is still able to achieve decent groundings, if no feedback is provided due to its unsupervised grounding mechanisms. Furthermore, the results for Scenario III also confirm that combined verbal and pointing feedback achieves both better grounding accuracy as well as sample efficiency than pointing-only feedback. In fact, the results show that too much pointing-only feedback can even lead to worse results than when no feedback is provided, which is not the case for Scenario II but highlights the benefit of combined verbal and pointing feedback. The next section (Section 4.5.4) will focus on answering the last research question regarding the robustness of the framework in regard to incorrect feedback.

4.5.4 Scenario III: Synthetic - Incorrect Feedback

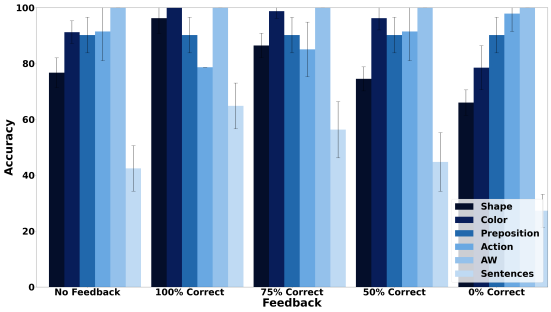
This section presents the results obtained for Scenario III (Section 3.5.3) for both feedback types when part of the provided feedback is incorrect to investigate the third research question, i.e. how incorrect feedback affects the grounding accuracy, for Scenario III and to verify the results obtained for Scenario II (Section 4.5.2).
Figure (4.11a) seems to indicate that even 100% incorrect pointing-only feedback provided every second situation leads still to an increase of grounding accuracy in comparison to the case when no feedback is provided. This observation seems at first counter-intuitive but becomes understandable when remembering that the pointing-only feedback mechanism maps all words in the utterance to the target object's color and shape CRs. Thus, even if the feedback is completely incorrect, it still strengthens the mappings from shape words to shape CRs and color words to color CRs. When combined pointing and verbal feedback is provided the grounding accuracy increases, even if 50% of the feedback is incorrect, independent of how often feedback is provided, i.e. every situation or every second situation. Interestingly, the accuracy of the action groundings increases with the percentage of incorrect feedback, while the accuracy of the shape and color groundings decreases.



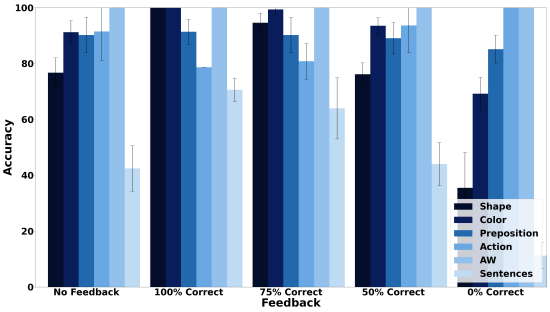
(a) Grounding results when pointing-only feedback is provided for 50% of the situations.



(b) Grounding results when pointing-only feedback is provided for all situations.

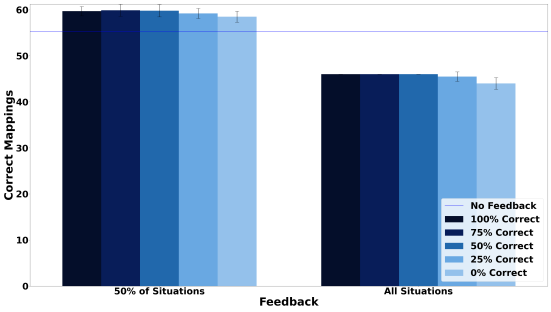


(c) Grounding results when combined pointing and verbal feedback is provided for 50% of the situations.

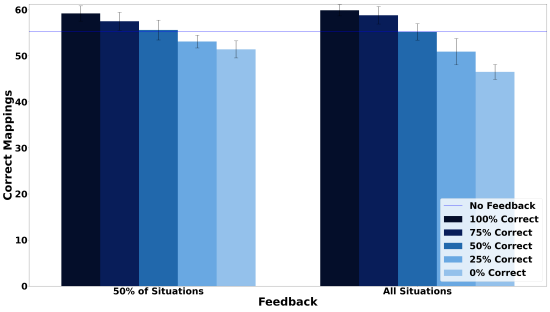


(d) Grounding results when combined pointing based and verbal feedback is provided for all situations.

Figure 4.11: Mean grounding accuracy results, corresponding standard deviations, and percentage of sentences for which all words were correctly grounded for Scenario III for both types of feedback and different rates of correct and wrong feedback.



(a) Pointing-only feedback.



(b) Pointing and verbal feedback.

Figure 4.12: Mean number and standard deviation of correct and false mappings over all 10,000 situations of Scenario III for different percentages of correct feedback, when feedback is provided for either 50% or all of the situations.

Figure (4.12a) confirms that pointing-only feedback always increases the number of correct mappings, if provided every second situation and always decreases the number

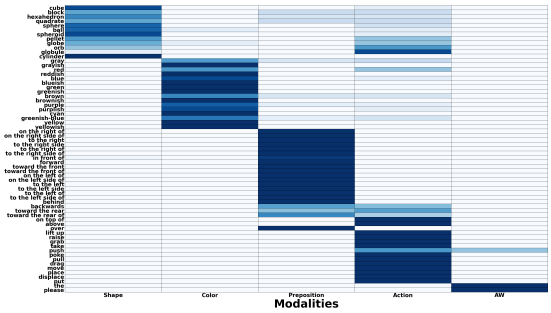
of correct mappings, if provided every situation. In contrast, when combined verbal and pointing feedback is provided, the results (Figure 4.12a) are similar to the results for Scenario II since the number of correct mappings is higher than when no feedback is provided, even if 50% of the feedback is incorrect, independent of how often feedback is provided. Thus, combined verbal and pointing feedback is more robust to incorrect feedback than pointing-only feedback.

Figure (4.13) illustrates which words, modalities, and CRs are mostly affected by incorrect feedback. For pointing-only feedback the highest increase in inter-modality confusion is for prepositions, which are most of the time wrongly mapped to shape CRs, while the confusion for shapes decreases. There exist only light intra-modality confusion for the shape word "globule", otherwise all confusion is across modalities (Figure 4.13d). For combined verbal and pointing feedback the highest increase in inter-modality confusion is for shapes and colors, while the confusion for prepositions decreases (Figure 4.13e). Colors are most of the time wrongly mapped to shape CRs, while shapes are most of the time wrongly mapped to action CRs and sometimes also to the CRs of prepositions. Figure (4.13f) shows that intra-modality confusion only exist for the word "globule", which is wrongly mapped to the CR of CYLINDER. Otherwise most confusion is across modalities.

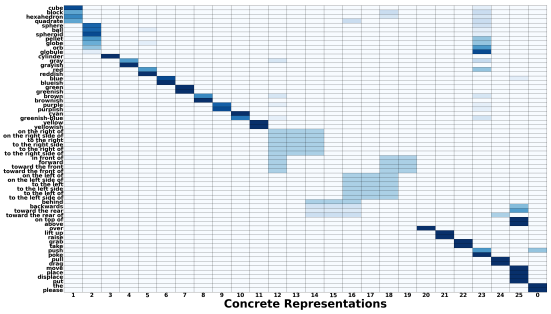
Overall the results show that the robustness of the framework in regard to incorrect feedback depends on the percentage of incorrect feedback. For example, if 50% of combined verbal and pointing feedback is incorrect when feedback is provided every situation it nearly negates the positive effect of the feedback. However, it is important to note that it is very unlikely that 50% of the provided feedback is incorrect, in fact, even 25% incorrect feedback is not very likely, if a robot is deployed in different environments and interacting with many different people. Thus, the results confirm the results obtained for Scenario II (Section 4.5.2), that the benefit of feedback outweights the potential damage incorrect feedback can cause.

4.6 Discussion

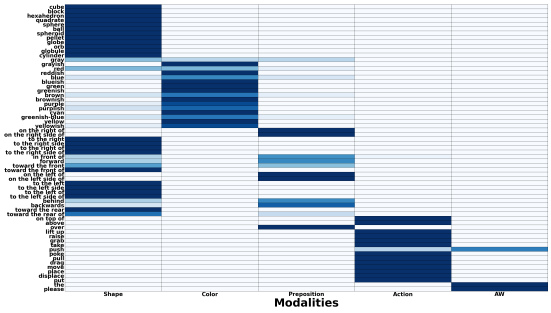
In this chapter a novel hybrid grounding framework was proposed (Section 4.3), which combines state-of-the-art unsupervised and supervised grounding mechanisms to combine the best of both paradigms, i.e. to achieve higher grounding accuracy and sample-efficiency through feedback without depending on it. This was achieved by integrating two feedback mechanisms into the unsupervised grounding framework proposed in Chapter (3) so that the framework is able to handle both pointing-only as well as combined verbal and pointing feedback. The proposed framework was evaluated through two different scenarios (Section 4.4) that differ based on the used modalities, words, CRs, and number of situations. The obtained results showed that both types of feed-



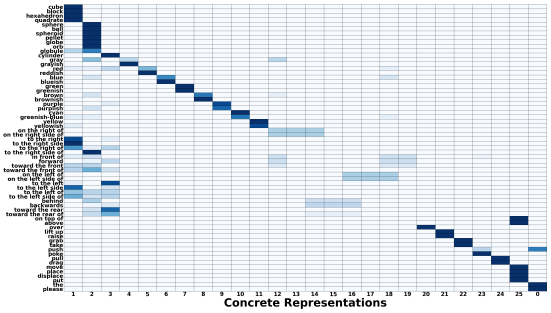
(a) Confusion matrix of words over different modalities when no feedback is provided.



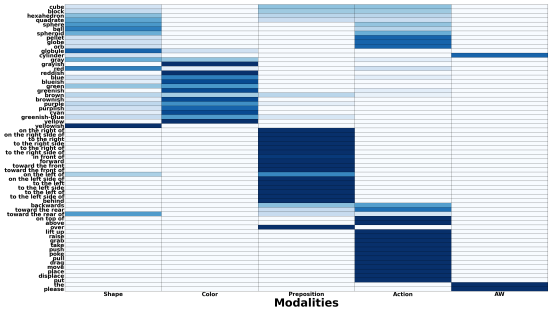
(b) Confusion matrix of words over different CRs when no feedback is provided.



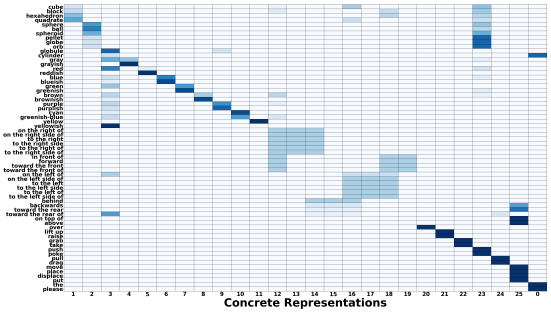
(c) Confusion matrix of words over different modalities when pointing-only feedback is provided for every situation.



(d) Confusion matrix of words over different CRs when pointing-only feedback is provided for every situation.



(e) Confusion matrix of words over different modalities when combined pointing and verbal feedback is provided for every situation.



(f) Confusion matrix of words over different CRs when combined pointing and verbal feedback is provided for every situation.

Figure 4.13: Confusion matrices for all ten situation sequences and three different types of interactions, i.e. no feedback, pointing-only feedback and combined verbal and pointing feedback, of Scenario III, when all feedback is incorrect.

back improve the accuracy of the obtained groundings and the sample-efficiency of the framework without preventing the framework to achieve decent grounding results when no feedback is available. Additionally, the results also showed that combined verbal and pointing feedback leads to more accurate groundings and a higher sample-efficiency than pointing-only feedback because the verbal feedback helps to make the mappings created due to the feedback more accurate.

Since it cannot be ensured that the provided feedback is always correct, it was also investigated how incorrect feedback influences the grounding performance. The investigation showed that the influence of incorrect feedback depends on the percentage of encountered incorrect feedback. For example, if 25% of pointing-only feedback is incorrect for the second scenario, the same grounding accuracy is achieved as if no feedback is provided, while for combined verbal and pointing feedback even 50% incorrect feedback provides still a benefit in terms of the number of correct mappings. This results illustrate that adding verbal feedback does not only improve the accuracy of the obtained groundings but makes the framework also more robust in regard to incorrect feedback. In summary this means that the benefit provided by feedback outweighs the possible damage by incorrect feedback, especially since it is unlikely that 25% of the encountered feedback will be incorrect, if a robot is employed in different environments and interacts with many different people because the majority of people will provide correct feedback.

In future work, the integration of other support mechanisms, like explicit teaching or demonstration, will be investigated, which might be useful to improve the grounding accuracy for actions or the accuracy for emotion types and intensities for Scenario IV since a user could teach or demonstrate different facial expressions and explain what emotions they represent. Another interesting point for future work is that the current feedback mechanisms only improve the groundings of shapes and colors, thus, it would be useful to modify or extend them to be able to also benefit other modalities.

5 Combining Language Grounding and Action Learning for Natural Task Learning

5.1 Motivation

Natural human-agent interaction requires agents to not only extract the meaning from natural language utterances but also to perform the tasks requested by humans, e.g. bringing a glass of water to a human. Due to the dynamicity of complex human-centered environments, it is impossible for an agent to learn all possible tasks in advance because minor changes in the environment might require the same task to be instantiated through different action sequences. Thus, agents need to learn the correct action sequence in an online fashion and even when executing the same task again, agents need to be able to adjust it to minor differences in the environment. Furthermore, learning must take place without explicit support from another agent because it can neither be assumed that another agent that is willing and able to provide the necessary support is always available nor can it be taken for granted that the provided support is appropriate and correct due to limited understanding by the other agent of how the artificial autonomous agent works, unintentional mistakes, or malicious intent to trick the artificial autonomous agent[1], which leads to the following research questions:

1. How can the goal state of a task be automatically extracted from natural language descriptions?

2. How can the agent ask for additional support, when it fails to extract the goal state?

3. How can the task be learned by the agent autonomously and without supervision, when only the goal state is known?

To answer above research questions, the grounding framework proposed in the previous chapter (Chapter 4) is extended with a RL based task learning mechanism to learn

[1] These are the same reasons that motivated the development of the unsupervised grounding mechanism presented and evaluated in Chapter (3).

tasks in an unsupervised manner using their goal states. Furthermore, a novel mechanism is proposed to extract goal states from natural language descriptions using previously obtained groundings. The rest of this chapter is structured as follows: Section (5.2) provides an overview of work related to task learning and goal state extraction from natural language. The proposed task learning and goal extraction mechanisms are explained in Section (5.3), while the employed scenarios and evaluation criteria are described in Section (5.4). Finally, Sections (5.5 and 5.6) present, evaluate and discuss the obtained results in regard to the investigated research questions, summarize the main contributions, describe limitations of the presented framework, and outline possible future work.

5.2 Related Work

Natural Task Learning requires two main research areas: learning of object manipulation tasks and grounding of actions and objects. The latter will enable the agent to identify both the objects involved in the task and the action that should be performed, while the former enables it to execute the requested action through corresponing actuator commands.

Many studies have investigated how object manipulation tasks can be automatically learned by robots, which usually requires a series of actions, i.e. actuator commands, to change the state or position of a target object [27]. Manipulation tasks are high-level macro actions that consist of sequences of low-level micro-actions, which can be defined in many different ways, thereby determining which learning approaches are most appropriate. For example, micro-actions can be represented through the movements of individual joints [35, 66], simple fine-grained movements of end effectors, or sophisticated and complex movements of end effectors or body parts, which allows the use of very high-level learning mechanisms, such as precise guidance through natural language instructions [88]. When micro-actions are represented through simple movements of joints or end-effectors, most studies employed learning through demonstration or RL [1, 36, 66, 100]. For the former, a human tutor has to demonstrate the desired task to the agent so that a policy can be derived from the recorded state-action pairs [6]. The latter, on the other hand, does not require the task to be demonstrated. Instead, it only requires a description of the goal state and discovers through trial-and-error possible policies [101].

Abdo et al. [1] proposed a method that enables robots to learn manipulation tasks, such as placing one object on another, from kinesthetic demonstrations, i.e. the robot's manipulator was manually moved by a human tutor to enable the robot to learn how to move its joints to perform the target task. Although only a small number of demonstrations was necessary to learn the tasks, requiring the manipulator to be directly moved

by a human tutor might not be possible in some situations and does not allow natural task learning as defined at the beginning of this chapter, i.e. the agent should be able to learn how to perform a task only by utilizing information about the goal of the task and without supervision.

Popov et al. [66] and Gudimella et al. [36] focused on learning to stack two objects onto each other through RL, by directly controlling the joints of a robotic arm and gripper, which led to high-dimensional action and state spaces requiring the experiments to be conducted in simulation due to the large number of required environment transitions. Although the employed models were able to learn the tasks based solely on the provided goal position, the studies simplified the tasks by always using the same goal position of the manipulation object with respect to the reference object. This is understandable because the main focus was on how to handle the high-dimensional action and state spaces, however, this is different from the focus of the study described in this chapter in which the main focus is not on how to learn macro-actions from lower-level micro-actions but how to learn and execute the correct lower-level action in repsonse to an instruction provided in natural language. Nevertheless, the use of RL meets the requirement to allow task learning without supervision when only the goal of the task is known.

While the studies described above have solely focused on the learning of manipulation tasks without considering grounding, most grounding studies that have investigated grounding of actions did not represent actions in a way that would allow their execution. For example, Taniguchi et al. [102] represented actions through a 38-dimensional vector that included information about the robot's posture, tactile information of the grapping hand, as well as the position of the hand relative to the target object. Thus, the employed action feature vector specified the goal position of the hand, which cannot be directly translated to actuator commands, unless a proper inverse kinematics solver exist for the employed robot. Salvi et al. [82] represented actions through simple symbols because the employed robot had been programmed to be able to perform the actions used in the study, which is not a realistic approach when considering that it is impossible to know in advance which tasks will be requested by another agent. Similarly, She et al. [88], Misra et al. [55], and She and Chai [87] already assumed that the employed robots were able to perform the lower-level actions through which the higher-level actions were grounded and did not consider learning them at the same time.

In contrast to the grounding studies mentioned above, Farkas et al. [23] proposed a model that was able to both learn and ground three actions in an experiment conducted using the iCub simulator [105]. The employed model consisted of three neural-network-based modules. The first neural network was used to detect the target object based on an input image of a tabletop scene with three objects, a 9 dimensional one-hot encoded color vector indicating the color of each object, and a six dimensional one-hot encoded

target vector indicating either the shape or color of the target object. The second module was used to learn the correct action sequence based on the provided action type and the position of the target object. Since states and actions were both continuous the Continuous Actor Critic Learning Automaton algorithm [107] was used to learn the correct action sequence. Finally, the third neural network was used to generate a linguistic description of the executed action. The main focus of the study was on the learning of the action, while the grounding process was only an additional component. Therefore, neither did the accuracy of the obtained groundings have an influence on the success of the action learning nor did the action learning have an influence on the accuracy of the obtained groundings. The same was the case for the studies presented in [75, 76] in which both grounding and action learning were investigated simultaneously without directly influencing each other.

In contrast to the majority of the aforementioned studies, the work presented in this chapter considers both grounding and action learning at the same time. Furthermore, in contrast to the few studies that did investigate both, the grounding and action learning components of the framework proposed in this study directly influence each other. More specifically, the obtained groundings are utilized to extract the goal states for the action learning component from natural language instruactions, while the action learning directly guides the grounding interaction, e.g. it influences when the agent will ask for feedback. Finally, although action learning is performed using RL, similar to [36, 66], the employed RL algorithm is much simpler due to the use of a relatively simple simulated environment leading to much smaller state and action spaces than the ones considered in [36, 66], although the target position of the object is not fixed.

Earlier versions of the action learning scenario have been published in [75, 76].The studies also considered simultaneous action learning and grounding, however, neither did the grounding results influence the action learning success nor did the action learning have any impact on the grounding performance. Furthermore, the employed grounding framework was an earlier version of the unsupervised grounding component of the grounding framework used in this chapter and was therefore not able to handle feedback from another agent. Finally, the perceptual information in the grounding scenario was synthetic like the perceptual information used in Scenario III (Section 5.4.2).

5.3 A Framework for Simultaneous Learning and Grounding of Actions

This section describes a novel framework that extends the hybrid grounding framework proposed in the previous chapter (Section 4.3) with a mechanism to learn how to execute the requested task described by the provided instruction using RL. To do this, the

framework utilizes previously obtained groundings to extract the goal state of the requested task from the provided natural language instruction. Furthermore, to increase the chance that the agent is able to extract the goal state, the framework allows to ask the external agent who gave the instruction for feedback regarding the determined target object, if the learning agent was able to determine a target object, or for a hint regarding the target object, if it was not able to determine it by itself.
Since both goal extraction and task clarification utilize previously obtained groundings, the accuracy of the obtained groundings has a direct influence on the action learning, while the success of the task learning has no direct influence on the grounding accuracy.
The proposed framework consists of 4 main parts: (1) Language grounding component (Section 5.3.1), which is able to determine word-CR mappings using both CSL and IL, (2) Goal extraction component (Section 5.3.2), which uses obtained groundings to determine the goal of the requested task, (3) Task clarification component (Section 5.3.3), which enables the agent to ask for support from the external agent, and (4) task learning component (Section 5.3.4), which uses RL to learn how to execute the requested task using the automatically extracted goal state. The inputs and outputs of the individual parts are highlighted below and in Figure (5.1), while they are described in detail in the following subsections.

1. **Language grounding component**:
   - **Input**: Natural language instructions, CRs, previously detected AWs, word and CR occurrence information, feedback information.
   - **Output**: Set of AWs and word to CR mappings.
2. **Goal extraction component**:
   - **Input**: Natural language instructions, CRs, AWs, and word-CR mappings.
   - **Output**: Goal state description.
3. **Task clarification component**:
   - **Input**: Natural language instructions and AWs.
   - **Output**: Word to CR mappings.
4. **Task learning component**:
   - **Input**: Situation and goal state descriptions.
   - **Output**: Learned task policy.

5.3.1 Language grounding

The language grounding component is the hybrid grounding framework proposed and evaluated in the previous chapter (Chapter 4). Thus, it uses both CSL and IL to deter-

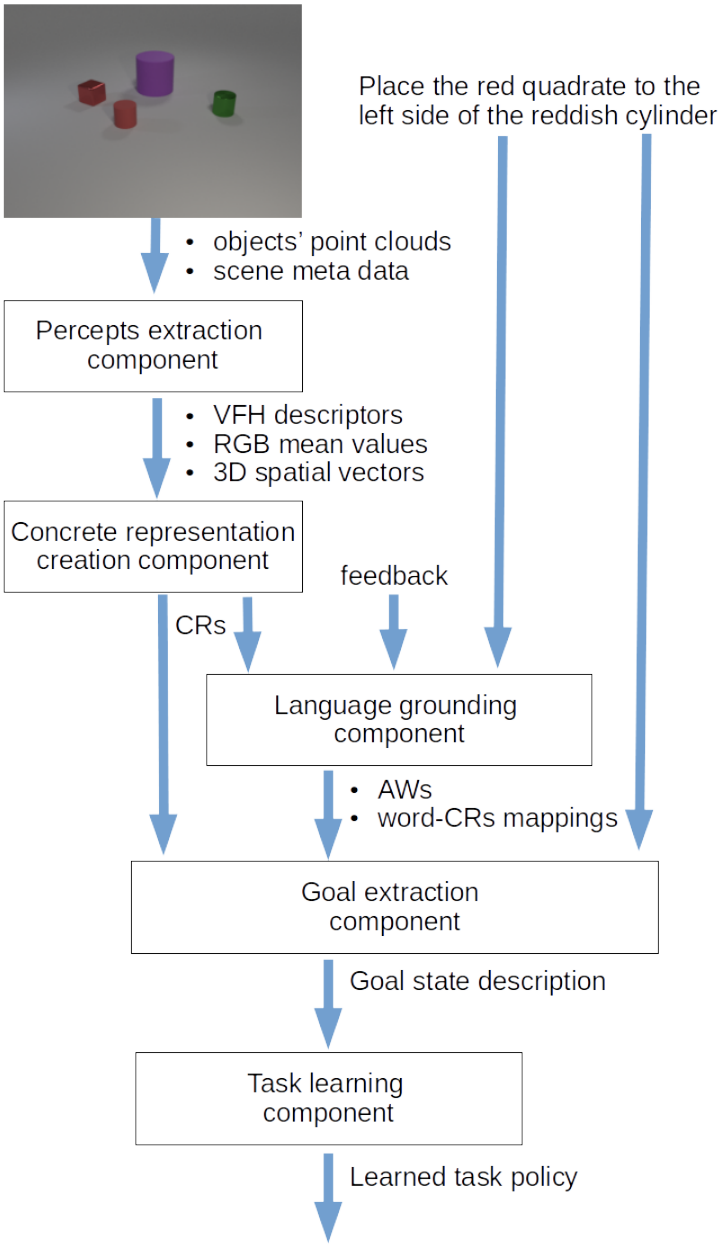


Figure 5.1: Illustration of the components of the proposed framework and the data flow for the second scenario (Section 3.5.2). First percepts, i.e. VFH descriptors, RGB mean values, and 3D spatial vectors, are extracted using the point clouds of the objects in the scene and the meta-data generated by the scene extraction script (see Section 3.5.2 for details). Afterwards, corresponding CRs are obtained, which are then provided as input to the language grounding and goal extraction components. Both components also take as input the natural language sentence. In addition, the language grounding component also receives any available feedback information as input, while the goal extraction component also receives as input the AWs and mappings obtained by the language grounding components. Finally, the goal state description is provided to the task learning component to learn the correct policy for the target task.

mine the correct mappings from words to CRs and is therefore able to utilize feedback provided by another agent, if available, without depending on it. When and how feedback is triggered and used by the agent is described in Section (5.3.3), while a detailed description of the grounding component itself is provided in Section (4.3) of the previous chapter.

5.3.2 Goal extraction

The goal extraction component tries to utilize previously obtained groundings to determine the goal state of the task requested in the current situation to provide it as input to the task learning component (Section 5.3.4). To extract the goal from the provided natural language instructions, the algorithm (Algorithm 8) checks for every non-AW through which CRs it is currently grounded and whether these CRs belong to an object, i.e. to the shape or color modality, or to the preposition modality. For this purpose, the agent utilizes the CR-modality map (CRM) and the set of detected objects (OBJ) created when the agent extracts the percepts from the environment of the current situation. The important thing to note is that the agent knows to which modality and in case of shapes and colors also which object a CR belongs, while it has no information about the modality of the words. The object that is referred to first will be selected as the target object (O_T) by the agent.
The idea behind this is that the target object is usually mentioned first in an instruction, which is the case for all instructions employed in the experiments described in this chapter. For example, "move the green cube to the left of the red cylinder" for Scenario II or "push the gray ball on the left side of the red cylinder in front of the green cube" for Scenario III. Thus, if the groundings are correct, the agent will manipulate the correct object. As a reference preposition (PRE_R), the agent will choose the last encountered preposition based on the fact that if two prepositions are encountered, the first preposition would describe the initial position of the target object, while the second preposition would describe the goal position in relation to the reference object. This again uses prior knowledge about the structure of the employed natural language instructions[2]. After all non-AWs have been processed, the reference object (O_R) is determined by looking at the number of determined prepositions and objects, i.e. P_{cr} and O_{obj}. Afterwards, the algorithm checks whether a target object (O_T) and reference preposition (P_R) have been determined and if that is the case, it provides them as well as the reference object (O_R), if available, to the task learning component (Section 5.3.4). Finally, the goal extraction

[2]While the utilization of prior knowledge about the structure of the instructions limits the applicability of the employed goal extraction mechanism to scenarios with similar instructions, it is similar to the selection or modification of a particular learning algorithm for a specific use case based on prior knowledge, e.g. the parameter settings for clustering algorithms, the network structure for neural networks, or the reward function in case of RL. Nevertheless, ways to make the goal extraction mechanism more general will be explored in future work.

Algorithm 8 The goal extraction procedure takes as input the words and CR of the current situation (W and CR), the set of detected AW (AWS), the set of previously obtained word-CR mappings ($WCRPS$), the set of current objects (OBJ) and prepositions (PRE), and the CR-modality map (CRM) and returns the target object (O_T), reference object (O_R), and the reference preposition (P_R).

procedure GOAL EXTRACTION($W, CR, AWS, WCRPS, OBJ, PRE, CRM$)
  $O_T = \{\}, O_R = \{\}, P_R = \{\}, O_{obj} = \{\}, P_{cr} = \{\}$
  for w in W **do**
    if $w \notin AWS$ **then**
      for cr in $WCRPS(w)$ **do**
        if $cr \in CR$ **then**
          for m in $CRM(cr)$ **do**
            if $m \in \{\text{shape}, \text{color}\}$ **then**
              for obj in OBJ **do**
                if $cr = obj_{cr}(m)$ **then**
                  if $O_T = \emptyset$ **then**
                    $O_T = obj$
                  if $obj \notin O_{obj}$ **then**
                    $O_{obj} = O_{obj} \cup obj$
            else if $m =$ preposition **then**
              for pre in PRE **do**
                if $cr = pre$ **then**
                  $P_{cr} = P_{cr} \cup \{cr\}$
                  if $|P_{cr}| = 2$ **then**
                    $P_R = PRE[1]$
                  else if $|P_{cr}| = 1$ **then**
                    $P_R = PRE[0]$
  if $|P_{cr}| = 2 \wedge |O_{obj}| \geq 2$ **then**
    $O_R = O_{obj}[2]$
  else if $|P_{cr}| = 1 \wedge |O_{obj}| \geq 1$ **then**
    $O_R = O_{obj}[1]$
  if $O_T \neq \emptyset \wedge P_R \neq \emptyset$ **then**
    Algorithm 10(O_T, O_R, P_R)
  return O_T, O_R, P_R

component returns the determined target object (O_T), reference object (O_R), and reference preposition (P_R) to the task clarification component (Section 5.3.3).

5.3.3 Task clarification

The task clarification component (Algorithm 9) sits above the other three components and is used to handle situations for which the goal extraction component (Section 5.3.2) fails to extract a goal state so that the agent is not able to learn the task without further information from another agent. First, it calls the grounding component (Algorithm 7) to update the set of current mappings ($WCRPS$ and $CRWPS$). Afterwards, it provides the updated mappings to the goal extraction component (Algorithm 8) to determine the current target object (O_T), reference object (O_R), and the reference preposition (P_R).
If the goal extraction component does not return a target object or reference preposition, the agent asks the external agent for help, which is provided in form of combined verbal and pointing feedback[3] (Algorithm 6). If the target object and reference preposition were returned, but the target object was wrong, the external agent provides also combined verbal and pointing feedback (Algorithm 6). Since it is very rare that repeated feedback will provide any benefit for the goal extraction mechanism, for each situation, the learning agent asks at most once for help and the external agent also provides at most once feedback, when the wrong object was manipulated.

5.3.4 Task learning

The task learning component is based on the RL component proposed by Roesler and Nowé [76], however, different from the original component used in [76], the agent needs to learn the correct action sequence in a single episode because trying multiple times does not seem plausible when considering deployment in human-centered environments. Unless the agent has an accurate simulation of the world so that it can try many times, i.e. for many episodes, in simulation to achieve the task and will only try to execute it in the real world, once it is able to do it reliable in simulation. The main challenge for the latter approach is the accurateness of the simulation to avoid that the task execution fails in reality after succeeding consistently in simulation.
Since both the state and action space are discrete, tabular Q-learning is used to find the optimal policy to reach the goal state as extracted from the natural language instruction. The Q-table is initialized with zeros. Since only one episode is used per task, the agent is not able to move the gripper or any of the objects out of the environment. The assump-

[3]The assumption for the experiment presented in this chapter is that the external agent will always provide support, when requested by the learning agent, and that the provided support is always correct. However, the results in the previous chapters (Sections 3.6 and 4.5) have shown that the framework does not require feedback and is able to learn the correct mappings over time in an unsupervised manner, and that the framework is also able to cope with wrong feedback.

Algorithm 9 The task clarification procedure takes as input all words and CRs of the instruction of the current situation (W and CR), the sets of previously obtained word-CR and CR-word pairs ($WCRPS$ and $CRWPS$), the set of previously detected AWs (AWS), the set of permanent phrases (PP, the sets of word and CR occurrences (WO and CRO), the set of permanent mappings (PMS), the set of detected objects (OBJ), the CR-modality map (CRM), and the sets of previously obtained word-CR and CR-word feedback ($WCRPSF$ and $CRWPSF$).

1: **procedure** TASK CLARIFICATION($W, CR, WCRPS, CRWPS, AWS, PP, WO, CRO, PMS, OBJ, CRM, WCRPSF, CRWPSF$)
2: $try = 1$
3: **while** $try \leq 3$ **do**
4: $WCRPS, CRWPS =$ *Algorithm 7*($W, CR, WCRPS, CRWPS, AWS, PP, WO, CRO, PMS$)
5: $O_T, O_R, P_R =$ *Algorithm 8*(W, CR, AWS, WCRPS, OBJ, CRM)
6: **if** $O_T == \emptyset \vee P_R == \emptyset$ **then**
7: Ask for support and receive verbal (WF) and pointing ($TOCR$) feedback
8: $WCRPSF, CRWPSF =$ *Algorithm 6*(W, WF, CR, TOCR, AWS, WCRPSF, CRWPSF)
9: **else if** O_T not correct **then**
10: Receive verbal (WF) and pointing ($TOCR$) feedback
11: $WCRPSF, CRWPSF =$ *Algorithm 6*(W, WF, CR, TOCR, AWS, WCRPSF, CRWPSF)
12: **else**
13: $try = 2$
14: $try+ = 1$

tion is that the agent has some form of hard-coded safety mechanism that intervenes, if the agent attempts an invalid action so that it will not be executed and the agent receives a reward of -1.

The observation vector provided to the agent contains the following information: (1) the shape of the manipulation object, (2) the gripper position relative to the manipulation object position, (3) the current manipulation object position relative to the current reference object position[4], and (4) gripper state, i.e. {open, closed}. Since the relative

[4]The reference object can be the same as the target object, if the agent is instructed to move the object relative to its initial position.

positions are used, the learned Q-table is applicable independent of the absolute object or gripper positions.

The agent can execute eight different actions, which are opening or closing the two-finger gripper, moving the gripper forwards, backwards, left or right, and lowering or raising the gripper. Physical interactions, e.g. when the gripper is moved to a position that is occupied by an object, are realistically simulated. This includes different behaviours based on the orientation of the gripper, the state of the gripper, and the shape of the object. For example, the object will be pushed by the fingers of the gripper when the gripper is moving to the right or left, or when the gripper is closed but not when the fingers are open and the gripper is moving forwards or backwards. Furthermore, balls will start rolling and will therefore move further than cubes. Thus, in the simulation, cubes are moved by one position and balls by two positions, unless an object occupies the second position, in which case the ball will also only be moved one position. Additionally, if the first position, to which the object is moved, is occupied by another object, both are moved.

For exploration ϵ-greedy is used as described by Sutton and Barto [101] and illustrated by lines 4-7 in Algorithm (10). The exploration rate ϵ is intially set to 0.4, i.e. 40% of the times the agent will select a random action, and decreases continuously with every situation[5] by a factor of 0.9999 so that the agent will execute many exploratory actions during the first situations but will focus more on exploiting accumulated knowledge for later situations.

When the manipulation object is placed on its goal position, the agent will receive a positive reward of 1. If the gripper or one of the objects is moved outside of the environment a negative reward of -1 is given. For each step a negative reward of -0.2 is given to encourage the agent to reach the goal state with the minimum number of possible steps. Additionally, potential-based reward shaping is used to reduce the number of suboptimal actions made and therefore the time required to learn [60]. The used Q-learning algorithm is represented by Equation (5.1), where a and a' are the actions taken in states s and s', respectively. α and γ represent the learning rate and discount factor, which are set to a value of 0.8 and 0.95, respectively. $F(s, s')$ is the potential-based reward, defined as the difference of the potential function ϕ over a source state s and destination state s' (Equation 5.2). For this study, the potential function ϕ is defined as illustrated by Equation (5.3), where P_R represents the reference preposition, G^{pos}, O_T^{pos}, and O_R^{pos} are the positions of the gripper, target object, and reference object, respectively, while s and s' represent the source and destination states of the current action.

An overview of the task learning procedure is provided by Algorithm (10). First, the Q-table is initialized with zeros. Afterwards, the agent checks whether the position of

[5]The exploration rate is actually decreased every episode, however, since only one episode is used per situation, it gets reduced every situation.

Algorithm 10 The task learning procedure takes as input the target object (O_T), reference object (O_R), and reference preposition (P_R).

```
1: procedure TASK LEARNING(O_T, O_R, P_R)
2:     Initialize Q-table
3:     while O_T^pos(s) ≠ O_R^pos(s) + P_R do
4:         if random_number < ϵ then
5:             Execute random action
6:         else
7:             Execute max(Q(s, :))
8:         Equation (5.1)
```

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + F(s,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{5.1}$$

$$F(s,s') = \gamma * \phi(s') - \phi(s) \tag{5.2}$$

$$\begin{aligned} \phi(s') &= \frac{1}{\|G^{pos}(s') - O_T^{pos}(s')\|_1 + \|O_T^{pos}(s') - O_R^{pos}(s') + P_R\|_1 + 1} \\ \phi(s) &= \frac{1}{\|G^{pos}(s) - O_T^{pos}(s)\|_1 + \|O_T^{pos}(s) - O_R^{pos}(s) + P_R\|_1 + 1} \end{aligned} \tag{5.3}$$

the target object is the same as the target position. If it is, the task learning procedure is terminated because the goal of the task has been achieved. Otherwise, the agent applies ϵ-greedy to determine whether it should exploit the previosuly learned knowledge encoded in the Q-table to select an action or explore the environment by selecting a random action. More specifically, it generates a random number between 0 and 1, and if the number is smaller than ϵ it will execute a random action to explore the environment, while otherwise, it will select the best action based on the information stored in the Q-table. Finally, after it executed the action, it updates the Q-table according to Equation (5.1).

5.4 Experiments

The proposed framework is evaluated through modified versions of the two scenarios used to evaluate the hybrid grounding framework in the previous chapter (Chapter 4).

Both scenarios are extended through a simulation of a task learning environment consisting of a tabletop with several objects and a robotic gripper. The idea is that a tutor provides an instruction from which the learning agent extracts the goal state of the requested task to learn through trial-and-error the correct action sequence. The employed experimental procedure is as follows.

1. Three or four objects are placed on a table and the agent determines the corresponding shape, color, preposition, and action[6] percepts.
2. A natural language instruction is given to the agent by a tutor and the agent uses CSL to update its groundings (Section 5.3.1).
3. The agent utilizes the updated mappings to extract the goal state from the instruction (Section 5.3.2).
4. (optional) The agent asks for help, if the goal state extraction failed and tries again. If it fails also the second time, the agent gives up and the experiment proceeds with the next situation (Section 5.3.3).
5. If the agent succeeded in extracting the goal state from the instruction, it learns how to reach the goal state using RL, thereby obtaining a corresponding action sequence (Section 5.3.4).
6. (optional) If the agent manipulated the wrong object, the tutor provides feedback about the goal state and task execution (Section 5.3.3).
7. (optional) The agent uses the feedback to improve its groundings (Section 5.3.1).

The tabletop environment is represented by a $7 \times 5 \times 2$ array so that positions are given as coordinates, i.e. $[x, y, z]$. Further details about the task learning environment are provided in Section (5.3.4), while the modifications applied to the two scenarios are described in detail in the following subsections.

5.4.1 Scenario II: CLEVR

The original Scenario II was slightly modified to be used in the task learning experiment because originally the employed utterances only described the current situation, i.e. the location of the target object, while for the task learning experiment they need to describe the desired position of the target object. The modification was done in three steps: (1) one of four action words, i.e. "move", "place", "displace", or "put", was appended to

[6]The action percepts are actually dummy percepts referring to the Q-table because for all actions the same Q-table is used and the task learning will be executed after the mappings for the current situation are obtained. Descriptions of the specific dummy action percepts used in the individual scenarios are provided in Sections (5.4.1 and 5.4.2).

Table 5.1: Overview of all concepts used in Scenario II with their corresponding synonyms and CR numbers (CR#) according to Figure (5.4).

Modality	Concept	Synonyms	CR#
Shape	CUBE	cube, block, hexahedron, quadrate	1
	SPHERE	sphere, ball, spheroid, pellet, globe, orb, globule	2
	CYLINDER	cylinder	3
Color	GRAY	gray, grayish	4
	RED	red, reddish	5
	BLUE	blue, blueish	6
	GREEN	green, greenish	7
	BROWN	brown, brownish	8
	PURPLE	purple, purplish	9
	CYAN	cyan, greenish-blue	10
	YELLOW	yellow, yellowish	11
Preposition	LEFT	to the left of, to the left side of	12, 13
	BEHIND	behind	13, 15
	FRONT	in front of	12, 14
	RIGHT	to the right of, to the right side of	14, 15
Action	MOVE	move, place, displace, put	16
Auxiliary Word	-	the	0

each sentence, (2) the preposition words were modified to describe the desired instead of the current position of the target object, e.g. "on the left of" was changed to "to the right of", and finally, (3) one action percept was added so that action words can be grounded through it. The action percept was represented through an one-hot encoded vector because the real *percept* is the Q-table containing the information how to perform the requested task. Table (5.1) illustrates the employed concepts, synonyms, and CRs for Scenario II.

5.4.2 Scenario III: Synthetic

In contrast to Scenario II, the third scenario already used sentences describing the desired action, thus, no modification to the sentences was necessary. However, action percepts were represented by five different one-hot encoded vectors, which seems unintuitive, when considering that all actions are in fact represented through the same Q-table. Therefore, at first, four of the five percepts were removed, but later they were re-added because it was discovered that better groundings are achieved, if the Q-table is represented through multiple percepts (see Section 5.5.2 for an explanation why this is the case). Table (5.2) illustrates the concepts, synonyms, and CRs used in Scenario III.

Table 5.2: Overview of all concepts used in Scenario III with their corresponding synonyms and CR numbers (CR#) according to Figure (5.7).

Modality	Concept	Synonyms	CR#
Shape	CUBE	cube, block, hexahedron, quadrate	1
	SPHERE	sphere, ball, spheroid, pellet, globe, orb, globule	2
	CYLINDER	cylinder	3
Color	GRAY	gray, grayish	4
	RED	red, reddish	5
	BLUE	blue, blueish	6
	GREEN	green, greenish	7
	BROWN	brown, brownish	8
	PURPLE	purple, purplish	9
	CYAN	cyan, greenish-blue	10
	YELLOW	yellow, yellowish	11
Preposition	LEFT	on the left of, on the left side of, to the left to the left side, to the left of, to the left side of	16, 17, 18
	BEHIND	behind, backwards, toward the rear, toward the rear of	14, 15, 16
	RIGHT	on the right of, on the right side of, to the right to the right side, to the right of, to the right side of	12, 13, 14
	FRONT	in front of, forward, toward the front, toward the front of	12, 18, 19
	ON	on top of, above, over	20
Action	LIFT UP, GRAB, PUSH, PULL, MOVE	lift up, raise, grab, take, push, poke, pull, drag, move, place, displace, put	21, 22, 23, 24, 25
Auxiliary Word	-	the	0
		please	

5.5 Results

In the following subsections the groundings obtained by the task learning framework (Section 5.3) for the two employed scenarios (Section 5.4) are presented and evaluated. The main questions investigated are (1) Whether the proposed goal extraction mechanism is able to automatically and accurately extract the goal state of the described task from natural language instructions, (2) Whether the extracted goal state is sufficient to learn all requested tasks, and (3) Whether the employed task learning procedure influences the accuracy of the obtained groundings. All three questions are investigated for both scenarios.

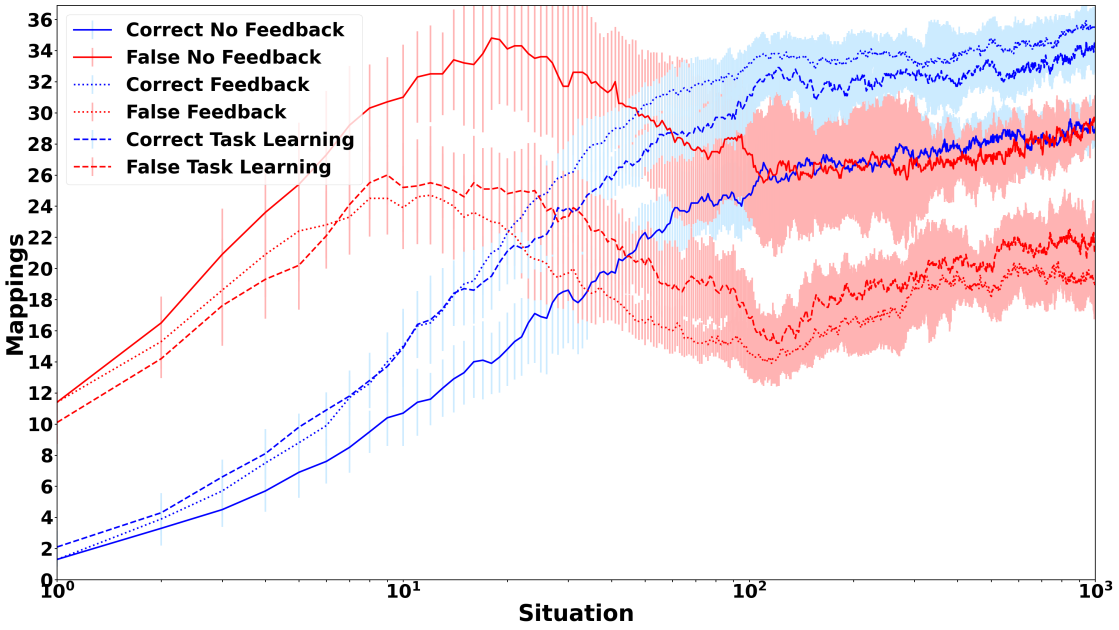


Figure 5.2: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 1,000 situations of Scenario II for all three investigated cases.

5.5.1 Scenario II: CLEVR

This section presents the grounding and task learning results for Scenario II (Section 5.4.1) and investigates the three research questions proposed at the beginning of the chapter. Figure (5.2) shows the number of correct and false mappings across all 1,000 situations of Scenario II for three different cases: (1) the case when no feedback is provided, (2) the case when combined verbal and pointing feedback is provided for every situation, which represents the best case for the agent, and finally (3) the case where the agent is also learning the task and receiving feedback, when asking for help or when the task was executed incorrectly, thereby, simulating a more realistic way of providing feedback.

For the first two cases no task learning was done, thus, the results are similar to the results presented in the previous chapters for Scenario II, except that the action modality has been added. When no feedback is provided, the framework is able to ground about 29 words correctly, while, if feedback is provided every situation or due to task learning, 35 and 34 correct mappings are obtained, respectively. This makes sense because feedback was provided during task learning for on average 83% of all situations and the results in Chapter (4) showed that the number of correct groundings increases with the amount of feedback provided in case of combined verbal and pointing feedback.

The accuracy results in Figure (5.3) overall confirm this, however, it is interesting that the accuracy for prepositions decreases, when more feedback is provided. The rea-

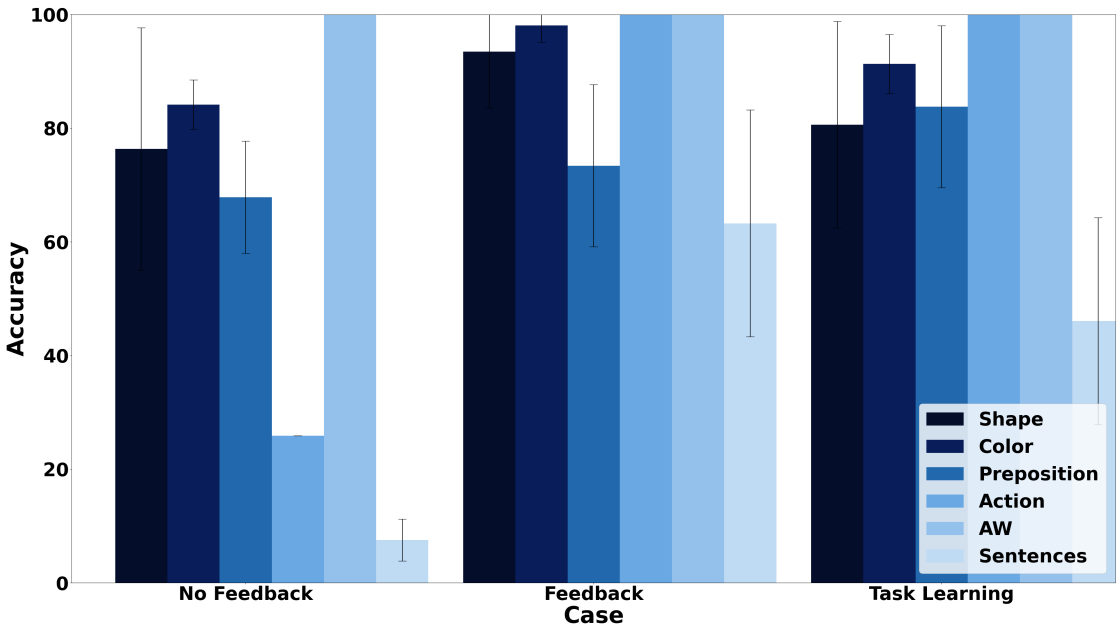
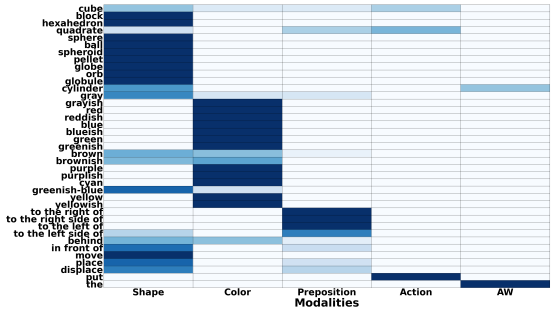


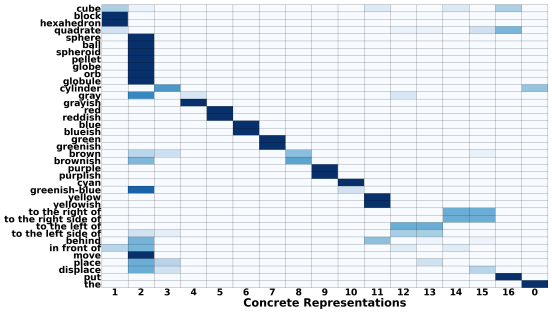
Figure 5.3: Mean grounding accuracy results and corresponding standard deviations for all modalities of Scenario II and all three investigated cases. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

son for this is that when feedback is provided every situation, the confusion between prepositions and actions increases (Figure 5.4), which was not the case for the results in Chapter (4) because there are no actions in the original Scenario II. Figure (5.4) shows that there is no intra-modality confusion, when feedback is provided and that the confusion for the third case, i.e. when feedback is provided due to the task clarification component, is only slightly higher than for the case when feedback is provided for all situations. The goal extraction component was able to extract a goal state for 26.9% of the situations and 88.1% of the determined target objects were correct. When the agent was able to extract the goal state, it was always able to learn how to reach it through RL. Overall, the results show that the proposed goal extraction mechanism is able to automatically extract the goal state of the described task from the provided natural language instructions. Especially, at the beginning, the extracted goal states are most of the time incorrect, however, this is not surprising because the mappings used to extract the goal state are also mostly incorrect, while during later situations the percentage of correct goal states increases. The results also show that the extracted goal state is sufficient for the task learning mechanism to perform the requested task as understood by the agent. Note that the extracted goal state was not always the correct goal state, but the RL algorithm was able to reach it anyway. Finally, the results show that the employed task learning procedure has an influence on the accuracy of the obtained groundings

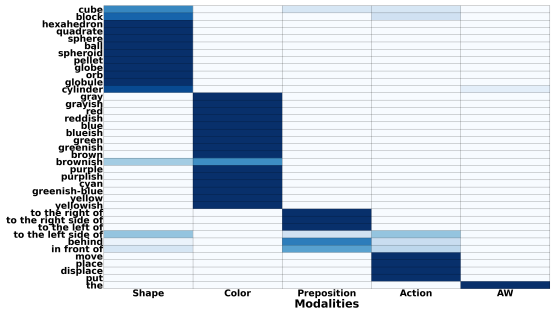
because it determines how much feedback the agent receives.



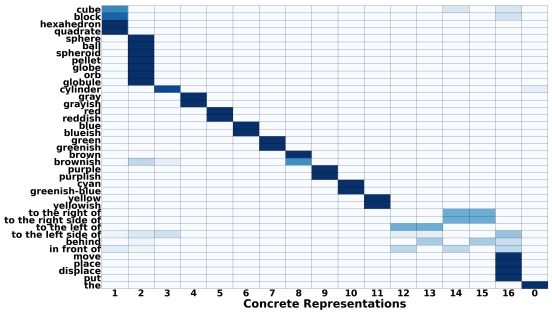
(a) Confusion matrix for the unsupervised framework proposed in Chapter (3).



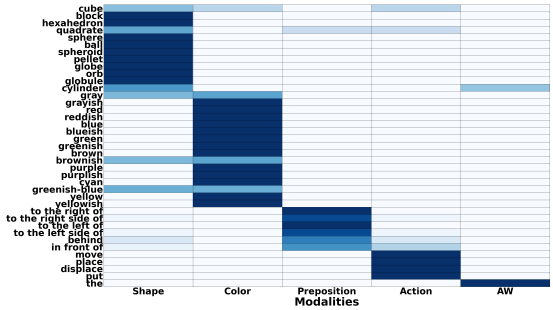
(b) Confusion matrix for the unsupervised framework proposed in Chapter (3).



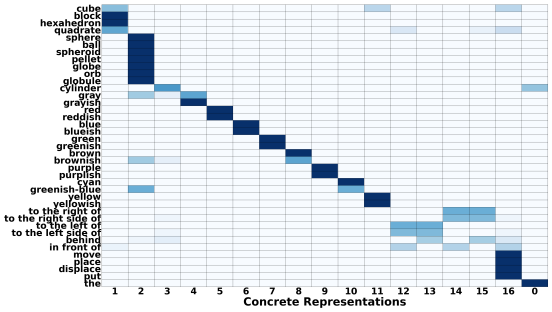
(c) Confusion matrix for the hybrid framework proposed in Chapter (4).



(d) Confusion matrix for the hybrid framework proposed in Chapter (4).



(e) Confusion matrix for the task learning framework proposed in this chapter.



(f) Confusion matrix for the task learning framework proposed in this chapter.

Figure 5.4: Confusion matrices showing how often each word of Scenario II was grounded through which modality and CR.

5.5.2 Scenario III: Synthetic

This section presents the grounding and task learning results for Scenario III (Section 5.4.2) and investigates the three research questions proposed at the beginning of the chapter. Figure (5.5) shows the number of correct and false mappings across all 10,000 situations of Scenario II for four different cases: (1) No feedback is provided, (2) Combined

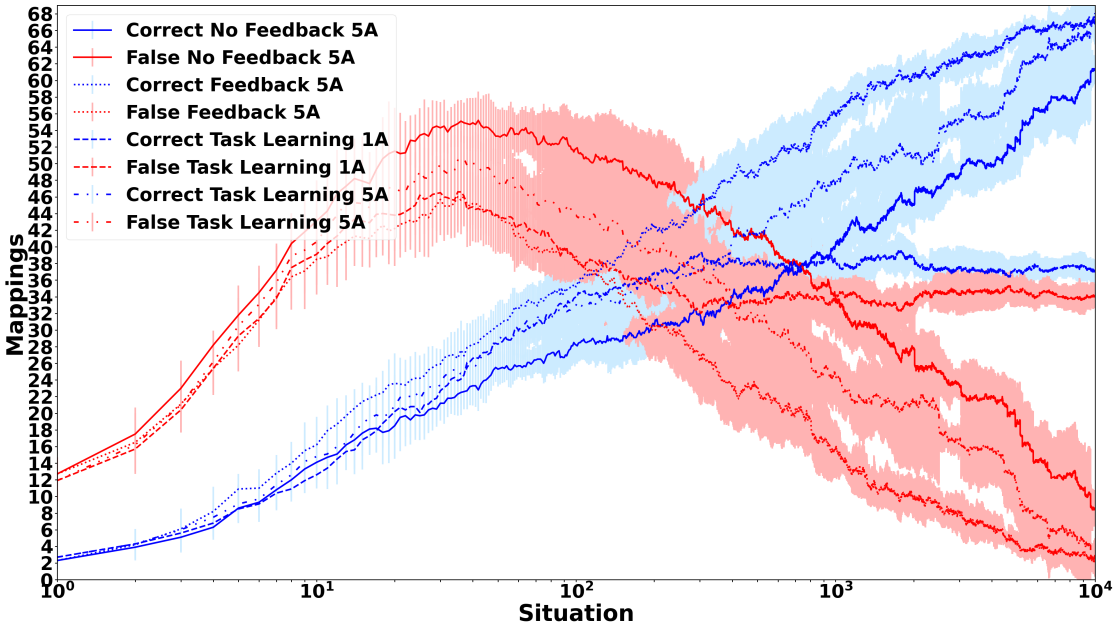


Figure 5.5: Mean number and standard deviation of correct and false mappings obtained by the proposed model over all 10,000 situations of Scenario III for all four investigated cases.

pointing and verbal feedback is provided for all situations, (3) Feedback is provided dynamically during the task learning interactions resulting in feedback for 65.5% of all situations and only one action percept and corresponding CR are used, and (4) Feedback is provided dynamically during the task learning interactions resulting in feedback for 76.5% of all situations while five action percepts and corresponding CRs are used.
During the first situations the number of correct and false mappings increases for all cases and the number of false mappings is much higher than the number of correct mappings. After about twelve situations the number of false mappings starts to decrease for all cases, while the number of correct mappings continues to increase. This trend continues for cases 1, 2, and 4 until the end, i.e. until all 10,000 situations have been encountered, while for the third case, i.e. when only one action percept is used, the number of correct and false mappings stagnates after about 200 situations so that the number of correct mappings stays at about 38 correct mappings. Although receiving feedback for every situation lets the number of correct mappings increase faster, after 10,000 situations the proposed task learning framework when using five action percepts has only one correct mapping less, which is reasonable when considering that feedback was provided for about 76.5% of all situations.
Figure (5.6) confirms that there is only a small difference between the case when feedback is provided for all situations or for 76.5% of the situations due to the dynamic interaction during task learning. The figure also nicely illustrates the importance of using

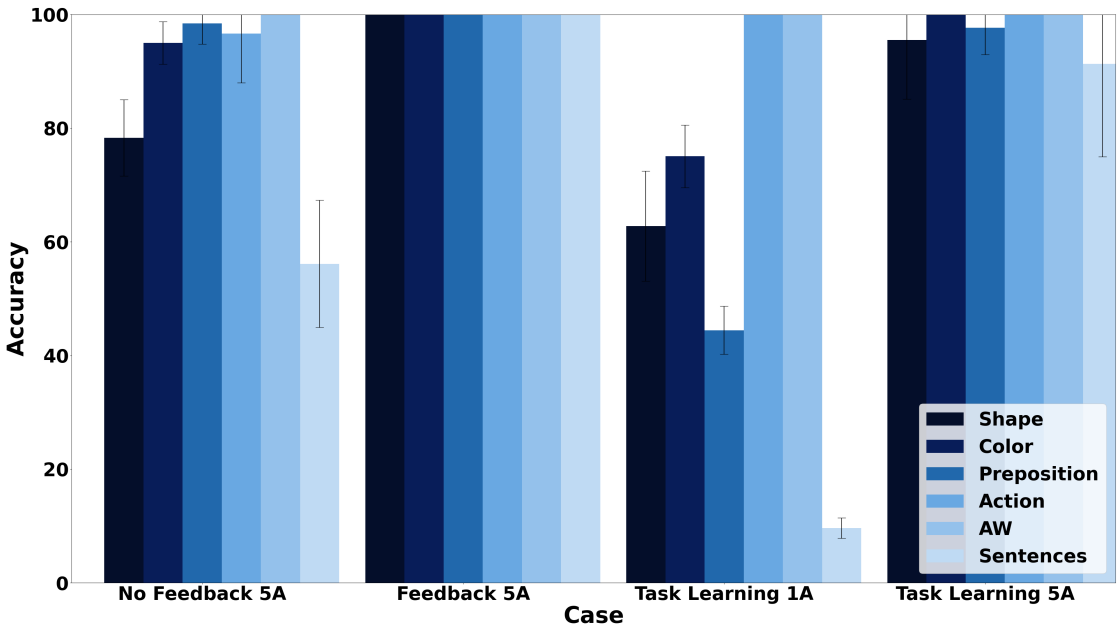
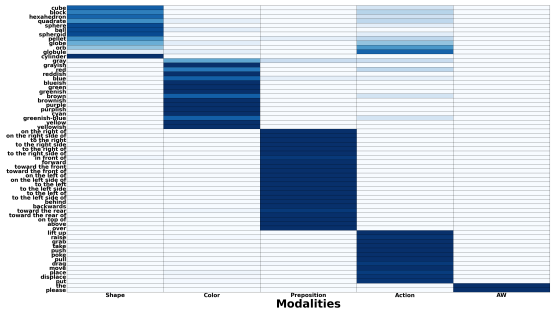


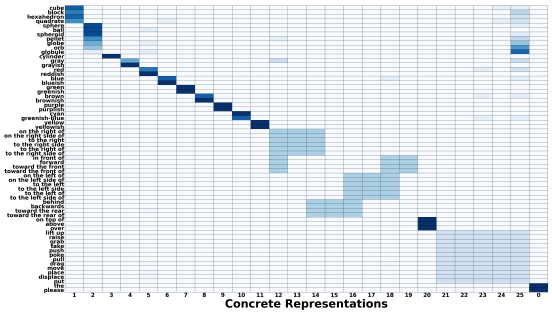
Figure 5.6: Mean grounding accuracy results and corresponding standard deviations for all modalities of Scenario III and all four investigated cases. Additionally, the percentage of sentences for which all words were correctly grounded is shown.

more than one action percept, although all of them represent the Q-table and are therefore not really different percepts, because the grounding accuracy for shapes, colors, and prepositions decreases strongly when using only one action percept. The reason is that most words are also at least partially grounded through the CR of the single action percept (Figure 5.7f) because it occurs in every situation while each action word only occurs on average in every twelfth situation. Using multiple action percepts does also not cause any harm since the action words will just be mapped to all of them equally (Figures 5.7b, 5.7d, and 5.7h). Every time the agent was able to extract a goal state from the natural language instruction, it was able to move the target object accordingly. The goal extraction component was able to extract a goal state for 42.4% of the situations and 85.3% of the determined target objects were correct.

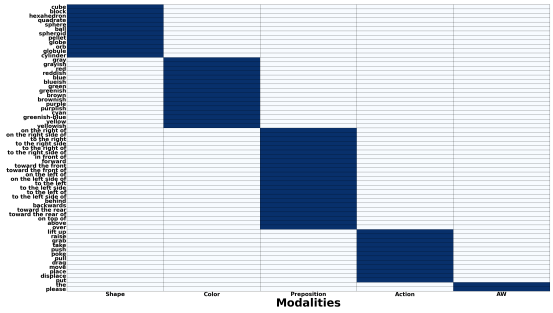
Overall, the results show that the task learning and clarification mechanisms have no negative impact on the groundings obtained by the framework, i.e. the accuracy of the obtained groundings is similar to the case, when feedback is provided for all situations, while they illustrate how previously obtained groundings can be used to extract the goal of the task automatically from language and how the agent can itself ask for support, if required.



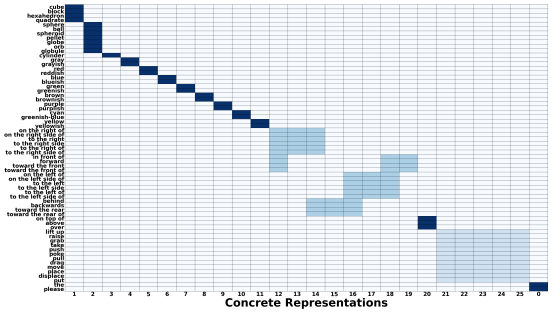
(a) Confusion matrix for the unsupervised framework proposed in Chapter (3), when five action percepts are used.



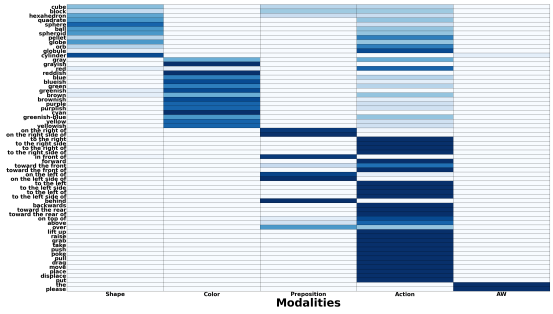
(b) Confusion matrix for the unsupervised framework proposed in Chapter (3), when five action percepts are used.



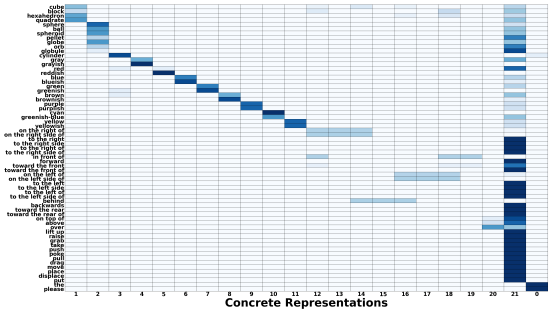
(c) Confusion matrix for the hybrid framework proposed in Chapter (4), when five action percepts are used.



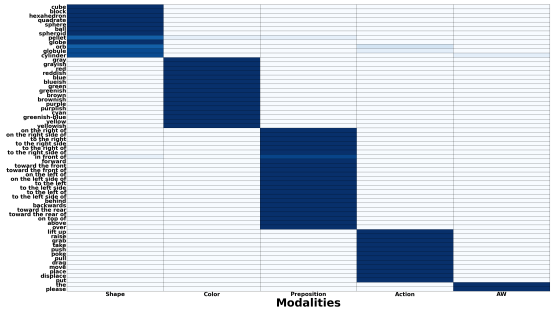
(d) Confusion matrix for the hybrid framework proposed in Chapter (4), when five action percepts are used.



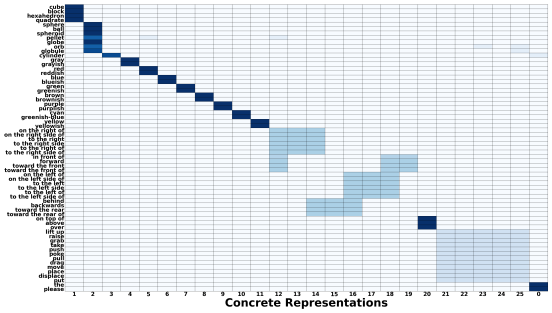
(e) Confusion matrix for the task learning framework proposed in this chapter, when one action percept is used.



(f) Confusion matrix for the task learning framework proposed in this chapter, when one action percept is used.



(g) Confusion matrix for the task learning framework proposed in this chapter, when five action percepts are used.



(h) Confusion matrix for the task learning framework proposed in this chapter, when five action percepts are used.

Figure 5.7: Confusion matrices showing how often each word of Scenario III was grounded through which modality and CR.

5.6 Discussion

Natural human-agent interaction requires agents to be able to perform the tasks requested by humans through natural language instructions, thus, agents do not only need to be able to understand the instructions but also to execute them. Therefore, this chapter proposed a framework for natural task learning, which utilized the hybrid grounding framework proposed in the previous chapter (Chapter 4) and combined it with mechanisms to (1) extract the goal states of the requested tasks from the corresponding natural language instructions, to (2) use the goal states as guidance for an RL algorithm to learn the tasks in a simulated grid-environment, and to (3) ask for support, when the goal extraction failed due to incorrect or noisy groundings.
The obtained results showed that the proposed mechanism can be used to extract the goal state from natural language instructions and that the success of the goal extraction depends on the accuracy of the obtained groundings. The results also showed that asking for support, when the goal extraction fails, is useful because it increases the accuracy of the groundings, which has a direct influence on the success of the goal extraction mechanism. Finally, the agent was able to learn the correct action sequences for all tasks for which goal states could be extracted using RL, thereby, confirming that RL can be used to let an agent learn a task autonomously, when only the goal state is known.
However, the task learning environment was relatively simple, thus, to learn the execution of a task in the real world a more sophisticated RL algorithm than tabular Q-learning must be used. Furthermore, the agent was not able to extract the goal state for about 75% of the encountered situations. While in the conducted experiment, it was no problem to just accept the failure of the agent to extract the goal state, when an agent would interact with real users, it would require some additional mechanisms to also be able to determine the goal states for these situations, e.g. by asking the user for a direct demonstration of the task. This is similar to the idea of the hybrid grounding framework to utilize as much support as possible, if available, without depending on it.
Therefore, in future work, the framework will be extended to also allow learning of the task execution from demonstration, if voluntarily provided by an external agent or provided on request from the learning agent in cases, when the goal state extraction from natural language failed. Furthermore, it will also be investigated how to automatically split complex tasks into subtasks to allow indirect grounding, i.e. grounding of more complex and higher level tasks through simpler lower level tasks that the agent might have already learned before, to simplify and speed up both the grounding as well as task learning process. This directly leads to another important part of future work, i.e. the explicit representation of concepts and integration of the task learning framework with a knowledge representation to explore the utilization of abstract knowledge to increase the sample-efficiency and accuracy of the grounding and task learning mech-

anisms as well as support the goal extraction from natural language. Finally, it will be interesting to investigate how to perform task learning in simulation to speed up the learning process and prevent the execution of potentially harmful or dangerous actions in the real world. Important for this is that the simulation is reasonably accurate so that the policies learned in simulation are useful in the real world.

6 Conclusions and Future Work

Artificial agents have the potential to not only make our lives more easy and comfortable by performing repetitive and physically demanding tasks but also to become important social partners by providing social, mental and physical care, especially for people in society that require constant support during their daily living, like elderly people, people with dementia, or people with disabilities. However, for this, artificial agents need to be able to interact with humans in a natural and efficient manner, which requires artificial agents to understand natural language because it is the main medium humans use to communicate. However, understanding natural language is non-trivial and requires that words and phrases are linked to the concepts they refer to and that the concepts are grounded in the physical world by linking them to corresponding CRs of percepts.
Since human-centered environments are complex and dynamic, grounding mechanisms must be able to obtain new groundings and update existing groundings in a continuous and open-ended fashion to account for changes in the environment and incorporate new information obtained through the agents' sensors. Additionally, natural interactions require artificial agents to be able to learn how to execute novel tasks, which they have never performed before without offline training or detailed supervision by a human tutor. However, agents should be able to learn from feedback or support provided by another agent, if available, without depending on it.

6.1 Summary of Presented Research

As a step towards a future in which artificial agents are able to interact in a natural manner with humans, this thesis introduced a novel task learning framework to enable agents to (1) continuously ground synonymous words and phrases through homonymous CRs without explicit assistance by a human tutor using CSL, (2) to utilize external support in form of verbal and pointing feedback to improve the sample-efficiency of the grounding mechanism and the accuracy of the obtained groundings without trusting or relying on the provided support, and (3) to utilize previously learned groundings to extract goal states from natural language instructions to enable learning of the correct execution of requested tasks through RL.

The unsupervised CSL based grounding component of the framework (proposed in Chapter 3) was evaluated through four different scenarios with different modalities, words, CRs, and number of situations. The obtained results showed that the unsupervised grounding component can be used to detect AWs and ground non-AWs and phrases through corresponding CRs in an unsupervised and open-ended manner, while outperforming a state-of-the-art Bayesian learning model based on the achieved grounding accuracy, online grounding capability, and sample-efficiency. In fact, the proposed framework is able to process new situations fast enough for real-world deployment, while this is not feasible with the baseline model due to its need for offline training and an overall much higher processing time. Additionally, the results also showed that the CSL based grounding component is able to handle language ambiguity in form of synonymy and homonymy.

Chapter (4) investigated whether the combination of CSL and IL through the integration of two feedback mechanisms into the unsupervised grounding framework could be used to improve its grounding accuracy and sample-efficiency, while ensuring that the framework still works when no or incorrect feedback is provided. The proposed hybrid framework and the two different feedback mechanisms, i.e. combined pointing and verbal feedback as well as pointing-only feedback, were evaluated through two of the four scenarios used to evaluate the unsupervised grounding component in Chapter (3). The obtained results showed that both types of feedback improve the accuracy of the obtained groundings and the sample-efficiency of the framework, while enabling the framework to still achieve decent grounding results, when no feedback is provided. Furthermore, the results also showed that combining verbal and pointing feedback leads to more accurate groundings and a higher sample-efficiency than if only pointing feedback is provided because the verbal feedback ensures that the influence of the feedback on the obtained mappings is more accurate.

Additionally, it was also investigated how robust the framework is to incorrect feedback, which is important since it cannot be assumed that the provided feedback is always correct. The results showed that the influence of incorrect feedback depends on the percentage of encountered incorrect feedback as well as the type of the provided feedback. For example, if 25% of pointing-only feedback is incorrect, the same grounding accuracy is achieved as if no feedback is provided for Scenario II, while for combined verbal and pointing feedback even 50% incorrect feedback still increases the number of correct mappings. This illustrates the benefit of combined verbal and pointing feedback because it does not only lead to more accurate groundings but increases also the robustness of the framework in regard to incorrect feedback. The results also highlight that the potential damage due to incorrect feedback is relatively limited, especially, since it is unlikely that 25% of the provided feedback is incorrect because most people that artificial agents would interact with would provide correct feedback. Thus,

the benefit of feedback clearly outweights the possible harm caused by incorrect feedback.

Empowering artificial agents to ground words and phrases through corresponding CRs independent of the availability of external support is essential but not sufficient, when aiming for natural and efficient human-agent interactions, because the artificial agents need to also be able to execute any tasks requested by a human. Therefore, Chapter (5) extended the hybrid grounding framework for natural task learning with mechanisms (1) to extract the goal states of requested tasks from the corresponding natural language instructions, (2) to use the goal states as guidance for an RL algorithm to learn the tasks in a simulated grid-environment, and (3) to ask for support, when the goal extraction failed due to incorrect or noisy groundings. The proposed natural task learning framework was evaluated through slightly modified versions of the two scenarios used to evaluate the hybrid grounding framework in Chapter (4).

The obtained results showed that the proposed mechanism can be used to extract goal states from natural language instructions and that the success of the goal extraction depends on the accuracy of the obtained groundings. The results also showed that asking for support in form of feedback, when the goal extraction fails, directly influences the success of the goal extraction mechanisms because it increases the accuracy of the obtained groundings. Finally, the agent was able to learn the correct action sequences for all tasks for which goal states could be extracted using RL, thereby, confirming that RL can be used to let agents learn tasks autonomously, when only their goal states are known.

However, the proposed framework and conducted experiments have several limitations leading to several avenues for future work, which are outlined in the next section.

6.2 Avenues for Future Work

One limitation of the results presented in Chapter (5) is that the task learning environment was represented through a small relatively simple grid-world, thus, the proposed RL algorithm cannot be directly deployed for task learning in the real world. Instead, to learn the execution of tasks in the real world a more sophisticated RL algorithm than tabular Q-learning must be used because the real world is (1) a continuous space with an infinite number of states that requires the use of function approximation, (2) stochastic so that the same action can lead to many different outcomes, and (3) dynamic due to the presence of other agents and natural forces, like wind, so that the environment is constantly changing, even if the agent is not performing any action. For example, enabling an artificial agent with a humanoid embodiment, i.e. not specifically designed for the requested task, to rake leaves in a garden will be very difficult because the agent must be able to handle a variety of tools, which is non-trivial when using tools designed

for the dexterity of the human hand, and needs to perform many low level actions to change the position of the leaves, which might sometimes also change independent of the actions of the agent due to external influences like wind.
Furthermore, the agent was not able to extract the goal state for about 75% of the encountered situations due to inaccurate and missing groundings. In the conducted experiment, this was not a problem because the agent could just give up and not perform the tasks for which it was not able to extract the goal state. However, this would in most cases not be a viable option when interacting with real human users who would expect the agent to be able to learn how to execute the task and if necessary, ask for assistance, e.g. in form of a demonstration, instead of just giving up. Thus, the framework needs to be extended with additional mechanisms to also be able to determine the goal states for these situations, e.g. by asking the user for a direct demonstration of the task. Important to note is that real world tasks, like raking leaves or cooking a meal, are much more difficult because they require multiple steps, which independently by themselves might already be non-trivial, e.g. grabbing a rake or a spoon.
Therefore, the framework will, in the future, not only be extended to allow learning of tasks from demonstration, thereby, following the same approach as for the grounding mechanism that it should be able to learn from support, if provided, without depending on it, but also to enable it to automatically split complex tasks into subtasks. The latter would have two main benefits. First, learning the execution of smaller tasks that require shorter action sequences to be performed is faster and easier because the reward signal will be less sparse and the risk that the agent never reaches the goal state will also be reduced. Second, splitting large tasks into subtasks will also allow indirect grounding, i.e. grounding of more complex and higher level tasks, like *bringing a glass of water*, through simpler lower level tasks, like *grabbing a glass* or *pouring water into a glass*, that the agent might have already learned before so that it can just perform the previously learned subtasks to instantly execute more complex and unknown tasks.
The idea of subtasks and indirect grounding automatically leads to another important part of future work, i.e. the explicit representation of concepts and integration of the task learning framework with a knowledge representation to explore the utilization of abstract knowledge to increase the sample-efficiency and accuracy of the grounding and task learning mechanisms as well as support the goal extraction from natural language. Furthermore, the use of a knowledge representation and therefore explicit concept representation allows the agent to reason about its actions as well as the input received via its sensors from the environment. This is crucial, when deploying an artificial embodied agent in complex human-centered environments because many tasks are potentially dangerous and having an automatically created and continuously updated model of the world will allow the agent to already restrict the set of actions that it will actually try out to the most plausible and least dangerous ones. Additionally, the ex-

plicit representation of concepts can also be utilized to benefit from the large amount of data available in written form, e.g. on the web or in books, to not only guide task learning but also help with the automatic detection of phrases and AWs since the purely unsupervised approaches used in this thesis and previous work are alone not sufficient, when considering the number of words and variations in sentence structure an agent would encountered during daily interactions with humans.
Overall, the framework proposed and evaluated in this thesis represents only a small step towards enabling artificial agents to flexibly learn new tasks to communicate and interact in a natural and efficient manner with humans or other agents in complex human-centered environments. While there are many things that can be investigated in future work as well as many possibilities to improve and extend the framework to handle more complex scenarios and achieve better grounding and task learning results, it is important to always keep in mind the big picture and final goal of enabling embodied agents to interact with humans in a natural and efficient manner in complex human-centered environments. Therefore, the next section will outline some of the bigger challenges that need to be addressed in future work.

6.3 Open Challenges

The development of frameworks and mechanisms (like the framework proposed in this study) that can be continuously improved and extended, instead of being only useful for one specific scenario or only used once for a specific study, is an important step toward to enable embodied agents to interact efficiently and naturally with humans in the real world. However, this also creates the danger of small incremental steps that focus not on achieving the original higher level goal but just on improving the previously developed frameworks. Incremental progress is important and to some degree necessary but only if it is moving in the right direction. Therefore, it is essential to evaluate the framework proposed in this thesis through interactions with humans in the real world. Initially, these interactions will not be without constraints because moving to the real world will immediately introduce additional challenges, which illustrates that language grounding and action learning are only two parts that are important but not sufficient to achieve natural and efficient interactions between humans and artificial agents. Following three important open challenges are described but there are many more challenges which are introduced by moving to interactions in complex human-centered environment.

1. One of the challenges introduced through unconstrained interactions in the real world is that agents need to be able to cope with noisy language data, which has not been considered in most grounding studies, i.e. in most grounding experi-

ments, like the ones presented in this thesis, only the perceptual information was noisy resulting in the use of CRs to extract the actual information from the noisy data, while the language data is usually presented as clear data without noise. However, this is not the case when considering deployment in the real world because most interactions with an embodied agent would be through speech and there is a certain probability that the noise in the speech signal will lead to incorrect conversion to text so that some words in the utterance provided to the grounding framework are incorrectly spelled or completely incorrect. If this happens only once a while it is fine, but if the speech to text conversion mechanism is not so accurate, it would cause substantial problems for unsupervised grounding mechanisms because they rely on co-occurrences of words and CRs. This problem would also occur, if humans interact in written form with the agent because it is very likely that the typed text will contain typos and some sentences might also not be grammatically correct.

2. Another challenge is that in daily speech many concepts are high-level and complex because they do not have a specific representation in the world that can be easily obtained with the sensors of the embodied agent. For example, how would a FINANCIAL BANK be represented? Through its building, which might look quiet different because some bank branches are in very old buildings, while other branches are in recently renovated or newly build buildings or through its logo, which might change over time? Or how should concepts like GOVERNMENT or PRESIDENT be grounded? Especially, since there are many different governments and many different presidents of many different countries and organizations so that it is non-trivial to decide what the correct mapping should be. In fact, the only solution is to allow a hierarchy of concepts so that the very general concept of PRESIDENT would be grounded through the lower level concepts of specific presidents, thereby, creating an abstract model of the world.

3. Finally, language is highly context dependent and most utterances assume that the hearer has commonsense and a basic understanding of the situation the speaker is in and the relationship between the speaker and hearer. Awareness of the situation does not need to be very high level, like understanding what to do when someones says “Can you give me a hand?”, which completely depends on the specific situation, but also includes understanding of pronouns in consecutive utterances. In the end, language is more than just a set of words or when grounded a set of concepts or CRs. Language is a medium to transfer a specific state in the mind or world model of one person to another person to change both their mind as well as in most cases also the state of the environment due to the action or inaction triggered by the transmitted information.

Glossary

Adjusted Rand Index (ARI) is a similarity measure between two clusters that has a value close to 0.0 for random labeling and exactly 1.0 when to clusters are identical [40].

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition leading to deficits in social communication and interaction [5].

Auxiliary Words (AW) are words or phrases that do not have CRs like articles, e.g. "a" or "the", and conjunctions, e.g. "and" or "as well as". While they do not have CRs they can still be essential for the meaning of an utterance, e.g. replacing the conjunction "neither...nor" with "both...and" reverses the meaning of the following utterance: "He neither shot the man nor threw his body into the river.".

Bidirectional Long Short-Term Memory (BiLSTM) is a sequence processing model consisting of one LSTM for each direction.

CMYK is a subtractive color model in which the subtractive primary colors (cyan, magenta, yellow) are added together to produce a wide array of colors. Additionally, black is used as a key plate because this produces better results than mixing cyan, magenta, and yellow to produce black.

Compositional Language and Elementary Visual Reasoning (CLEVR) is a synthetic dataset for visual question answering containing images of 3D-rendered objects [41].

Concrete Representations (CRs) represent sets of invariant features that are sufficient to distinguish perceptual and actuator information belonging to different concepts [37] and can be obtained through any clustering or classification algorithm.

Cross-Situational Learning (CSL) is a mechanism for word learning that is able to handle referential uncertainty by learning the meaning of words across multiple exposures [90] (Section 2.3).

Interactive Learning (IL) in the area of language grounding refers to supervised approaches in which the language learner receives support and feedback, e.g. pointing or eye gaze, from a tutor (Section 2.4).

Long Short-Term Memory (LSTM) is an artificial neural network with feedback connections that allow it to process sequences of data [38].

Markov Decision Process (MDP) is a mathematical framework for modelling decision making in discrete, stochastic, and sequential environments [47].

Perfect Emotion Types (PERT) refers to the case in Scenario IV where perfect concrete representations are used for emotion types to investigate the effect of the accuracy of the concrete representations on the grounding performance.

Predicted Emotion Types (PRET) refers to the case in Scenario IV where predicted concrete representations are used for emotion types, which is the normal case, but different from Perfect Emotion Types (PERT) that was used to investigate the effect of the accuracy of concrete representations on the grounding performance.

Rectified Linear Unit (ReLU) is an activation function used in deep learning that returns 0, when the input is negative, and the actual value, if the input is positive.

Reinforcement Learning (RL) is a framework that allows artificial agents to learn how to act in a correct and optimal manner in a complex environment through the maximization of a reward signal [101] (Section 2.6).

RGB is an additive color model in which the primary colors (red, green, blue) are added together to produce a wide array of colors.

Set of Auxiliary Words (AWS) is the set of detected auxiliary words, which is employed by the proposed framework to remove auxiliary words prior to grounding. Section (3.3.2) describes the auxiliary word detection procedure..

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is a multimodal dataset of emotional speech and song consisting of vocalized lexically-matched statements of 24 professional actors. Speech includes seven emotions, i.e. calm, happy, sad, angry, fearful, surprise, and disgust, at two levels of emotional intensity and neutral, while song includes five emotions, i.e. calm, happy, sad, angry, and fearful at two levels of emotional intensity and neutral [48].

Train/Test Split 60 (TTS60) refers to the case where only 60% of situations are used for training and the remaining 40% for testing to investigate how well models perform for unseen situations. For models that are able to continuously learn, it introduces an artificial and unnecessary limitation by deactivating its learning mechanism for 40% of the encountered situations.

Train/Test Split 100 (TTS100) refers to the case where all situations are used for training and testing to ensure that online learning models, i.e. models that are able to learn continuously, which requires an unrealistic benefit for models that require an explicit offline training phase.

Viewpoint Feature Histogram (VFH) is a point cloud descriptor representing the geometry of an object taking into account the viewpoint while ignoring scale variance [81].

Bibliography

[1] N. Abdo, L. Spinello, W. Burgard, and C. Stachniss. Inferring what to imitate in manipulation actions by using a recommender system. In *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[2] N. Akhtar and L. Montague. Early lexical acquisition: the role of cross-situational learning. *First Language*, 19(57):347–358, September 1999.

[3] A. Aly and T. Taniguchi. Towards understanding object-directed actions: A generative model for grounding syntactic categories of speech through visual perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018.

[4] A. Aly, A. Taniguchi, and T. Taniguchi. A generative framework for multimodal learning of spatial concepts and object categories: An unsupervised part-of-speech tagging and 3D visual perception based approach. In *IEEE International Conference on Development and Learning and the International Conference on Epigenetic Robotics (ICDL-EpiRob)*, Lisbon, Portugal, September 2017.

[5] American Psychiatric Association and others. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5®)*. American Psychiatric Pub, 5th edition edition, 2013.

[6] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57:469–483, 2009.

[7] E. Bagheri, O. Roesler, H.-L. Cao, and B. Vanderborght. Emotion intensity and gender detection via speech and facial expressions. In *Proceedings of the 31th Benelux Conference on Artificial Intelligence (BNAIC)*, Leiden, The Netherlands, November 2020.

[8] R. Bedford, T. Gliga, K. Frame, K. Hurdy, S. Chandler, M. Johnson, and T. Charman. Failure to learn from feedback underlies word learning difficulties in toddlers at risk for autism. *Journal of Child Language*, 40(1):29–46, 2013. doi: 10.1017/S0305000912000086.

[9] T. Belpaeme and A. Morse. Word and category learning in a continuous semantic domain: Comparing cross-situational and interactive learning. *Advances in Complex Systems*, 15(3 & 4), March 2012.

[10] J. Bleys, M. Loetzsch, M. Spranger, and L. Steels. The grounded color naming game. In *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009.

[11] P. Bloom. Précis of how children learn the meanings of words. *Behavioral and Brain Sciences*, 24:1095–1103, 2001.

[12] R. A. Blythe, K. Smith, and A. D. M. Smith. Learning times for large lexicons through cross-situational learning. *Cognitive Science*, 34:620–642, 2010.

[13] M. Cakmak, C. Chao, and A. L. Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, June 2010.

[14] S. Carey. The child as word-learner. In M. Halle, J. Bresnan, and G. A. Miller, editors, *Linguistic theory and psychological reality*, pages 265–293. MIT Press, Cambridge, MA, 1978.

[15] S. Carey and E. Bartlett. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17–29, 1978.

[16] G. Chevalier, S. T. Sinatra, J. L. Oschman, K. Sokal, and P. Soka. Earthing: Health implications of reconnecting the human bodyto the earth's surface electrons. *Journal of Environmental and Public Health*, 2012(29154), January 2012.

[17] E. V. Clark. The principle of contrast: A constraint on language acquisition. In *Mechanisms of Language Acquisition*, pages 1–33. Lawrence Erlbaum Associates, 1987.

[18] H. H. Clark and S. E. Brennan. Grounding in communication. In R. M. Baecker, editor, *Groupware and computer-supported cooperative work: Assisting human-human collaboration*, pages 222–233. Morgan Kaufman Publishers, Inc., 1991.

[19] C. Craye, D. Filliat, and J.-F. Goudou. Environment exploration for object-based visual saliency learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[20] C. R. Dawson, J. Wright, A. Rebguns, M. V. Escárcega, D. Fried, and P. R. Cohen. A generative probabilistic framework for learning spatial language. In *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, Osaka, Japan, August 2013.

[21] Oxford English Dictionary. Updates to the OED. URL https://public.oed.com/updates/. Accessed: 15/05/2021.

[22] F. Eyben, M. Wöllmer, and B. Schuller. Openear - introducing the munich open-source emotion and affect recognition toolkit. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, Amsterdam, Netherlands, September 2009.

[23] I. Farkas, T. Malík, and K. Rebrová. Grounding the meanings in sensorimotor behavior using reinforcement learning. *Frontiers in Neurorobotics*, 6, February 2012.

[24] S. Filin and N. Pfeifer. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry & Remote Sensing (P&RS)*, 60:71–80, 2006.

[25] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (CACM)*, 24(6):381–395, June 1981.

[26] C. Fisher, D. G. Hall, S. Rakowitz, and L. Gleitman. When it is better to receive than to give: Syntactic and conceptual constraints on vocabulary growth. *Lingua*, 92:333–375, 1994.

[27] R. Flanagan, M. C. Bowman, and R. S. Johansson. Control strategies in object manipulation tasks. *Current Opinion in Neurobiology*, 16:650–659, 2006.

[28] J. F. Fontanari and L. I. Perlovsky. Language acquisition and category discrimination in the modeling field theory framework. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1180–1185, Orlando, USA, August 2007. doi: https://doi.org/10.1109/IJCNN.2007.4371125.

[29] J. F. Fontanari and L. I. Perlovsky. How language can help discrimination in the neural modelling fields framework. *Neural Networks*, 21(2-3):250–256, March-April 2008.

[30] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, R. Ilin, and L. I. Perlovsky. Cross-situational learning of object-word mapping using neural modeling fields. *Neural Networks*, 22(5-6):579–585, July-August 2009.

[31] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, and L. I. Perlovsky. A cross-situational algorithm for learning a lexicon using neural modeling fields. In *International Joint Conference on Neural Networks (IJCNN)*, Atlanta, GA, USA, June 2009.

[32] M. C. Frank, N. D. Goodman, and J. B. Tenenbaum. A bayesian framework for cross-situational word-learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS)*, pages 457–464, Vancouver, Canada, December 2007.

[33] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6(6):721–741, November 1984.

[34] J. Gillette, H. Gleitman, L. Gleitman, and A. Lederer. Human simulations of vocabulary learning. *Cognition*, 73:135–176, 1999.

[35] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May-June 2017.

[36] A. Gudimella, R. Story, M. Shaker, R. Kong, M. Brown, V. Shnayder, and M. Campos. Deep reinforcement learning for dexterous manipulation with concept networks. *CoRR*, 2017. URL http://arxiv.org/abs/1709.06977.

[37] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[38] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. doi: https://doi.org/10.1162/neco.1997.9.8.1735.

[39] J. S. Horst and L. K. Samuelson. Fast mapping but poor retention by 24-month-old infants. *Infancy*, 13(2):128–157, February 2010. doi: 10.1080/1525O00070179559.

[40] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, December 1985. doi: https://doi.org/10.1007/BF01908075.

[41] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, July 2017.

[42] C. C. Kemp, A. Edsinger, and E. Torres-Jara. Challenges for robot manipulation in human environments. *IEEE Robotics & Automation Magazine*, 14(1):20–29, March 2007.

[43] D. P. Kingma and L. J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, USA, May 2015.

[44] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Osaka, Japan, March 2010.

[45] K. Koster and M. Spann. Mir: An approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(5):430–444, May 2000.

[46] P. A. Levine. *Healing Trauma*. Sounds True, Inc., Boulder, CO, 2008.

[47] M. L. Littman. Markov decision processes. In N. J. Smelser and P. B. Baltes, editors, *International Encyclopedia of the Social & Behavioral Sciences*, pages 9240–9242. Pergamon, Oxford, 2001.

[48] S. R. Livingstone and F. A. Russo. The ryerson audio-visual database of emotional speechand song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLoS ONE*, 13(5), May 2018.

[49] L. S. Lopes and A. Chauhan. How many words can my robot learn?: An approach and experiments with one-class learning. *Interaction Studies*, 8(1):53–81, April 2007.

[50] E. Margolis and S. Laurence. The ontology of concepts - abstract objects or mental representations? *Nous*, 41(4):561–593, October 2007.

[51] D. Marocco, A. Cangelosi, K. Fischer, and T. Belpaeme. Grounding action words in the sensorimotor interaction with the world: Experiments with a simulated icub humanoid robot. *Frontiers in Neurorobotics*, 4, May 2010.

[52] Merriam-Webster. We added new words to the dictionary for january 2021. URL https://www.merriam-webster.com/words-at-play/new-words-in-the-dictionary. Accessed: 15/05/2021.

[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ArXiv e-prints*, January 2013. eprint: 1301.3781.

[54] T. Mikolov, W. t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013.

[55] D. K. Misra, K. Tao, P. Liang, and A. Saxena. Environment-driven lexicon induction for high-level instructions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 992–1002, Beijing, China, July 2015.

[56] D. K. Misra, J. Sung, K. Lee, and A. Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *International Journal of Robotics Research (ijrr)*, 35(1-3):281–300, January 2016. doi: 10.1177/0278364915602060.

[57] T. Nakamura, T. Nagai, and N. Iwahashi. Grounding of word meanings in multimodal concepts using LDA. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, October 2009.

[58] J. Nevens and M. Spranger. Computational models of tutor feedback in language acquisition. In *7th Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, Lisbon, Portugal, September 2017.

[59] J. Nevens, P. Van Eecke, and K. Beuls. From continuous observations to symbolic concepts: A discrimination-based strategy for grounded concept learning. *Frontiers in Robotics and AI*, 7, June 2020.

[60] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In I. Bratko and S. Dzeroski, editors, *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, volume 99, pages 278–287, 1999.

[61] A. Nguyen and B. Le. 3D point cloud segmentation: A survey. In *6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Manila, Philippines, November 2013. IEEE.

[62] T. Nomura, T. Kanda, H. Kidokoro, Y. Suehiro, and S. Yamada. Why do children abuse robots? *Interaction Studies*, 17(3):348–370, January 2016.

[63] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 727–734, San Francisco, CA, June-July 2000.

[64] J. C. Pfeiffer. Principles of electrical grounding. Technical report, Pfeiffer Engineering Co., Inc., 2001.

[65] S. Pinker. *Learnability and cognition*. MIT Press, Cambridge, MA, 1989.

[66] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller. Data-efficient deep reinforcement learning for dexterous manipulation. *CoRR*, 2017. URL http://arxiv.org/abs/1704.03073.

[67] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978. doi: 0.1017/S0140525X00076512.

[68] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., New York, NY, USA, 1994.

[69] O. Roesler. A cross-situational learning based framework for grounding of synonyms in human-robot interactions. In *Proceedings of the Fourth Iberian Robotics Conference (ROBOT)*, Porto, Portugal, November 2019.

[70] O. Roesler. Enhancing unsupervised natural language grounding through explicit teaching. In *Proceedings of the UKRAS20 Conference: "Robots into the real world"*, Lincoln, UK, April 2020.

[71] O. Roesler. Enhancing unsupervised language grounding through online learning. In *ICRA 2020 Workshop "Shared Autonomy: Learning and Control"*, Paris, France, June 2020.

[72] O. Roesler. Unsupervised online grounding of natural language during human-robot interaction. In *Second Grand Challenge and Workshop on Multimodal Language at ACL 2020*, Seattle, USA, July 2020.

[73] O. Roesler. Combining unsupervised and supervised learning for sample efficient continuous language grounding. *Frontiers in Robotics and AI*, 9, September 2022. doi: https://doi.org/10.3389/frobt.2022.701250.

[74] O. Roesler and E. Bagheri. Unsupervised online grounding for social robots. *Robotics*, 10(2), April 2021. doi: https://doi.org/10.3390/robotics10020066.

[75] O. Roesler and A. Nowé. Simultaneous action learning and grounding through reinforcement and cross-situational learning. In *ALA 2018, Adaptive Learning Agents Workshop.*, Stockholm, Sweden, July 2018.

[76] O. Roesler and A. Nowé. Action learning and grounding in simulated human robot interactions. *The Knowledge Engineering Review*, 34(E13), November 2019.

[77] O. Roesler, A. Aly, T. Taniguchi, and Y. Hayashi. A probabilistic framework for comparing syntactic and semantic grounding of synonyms through cross-situational learning. In *ICRA-18 Workshop on Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding.*, Brisbane, Australia, May 2018.

[78] O. Roesler, A. Aly, T. Taniguchi, and Y. Hayashi. Evaluation of word representations in grounding natural language instructions through computational human-robot interaction. In *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 307–316, Daegu, South Korea, March 2019.

[79] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, 3rd edition edition, 2010.

[80] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (pcl). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[81] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3D recognition and pose using the viewpoint feature histogram. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2155–2162, Taipei, Taiwan, October 2010.

[82] G. Salvi, L. Montesano, A. Bernardino, and J. Santos-Victor. Language bootstrapping: Learning word meanings from perception - action association. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(3):660–671, June 2012.

[83] A. D. Sappa and M. Devy. Fast range image segmentation by an edge detection strategy. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling (3DIM)*, Quebec City, Quebec, Canada, August 2002.

[84] J. Schaffer. On what grounds what. In D. Chalmers, D. Manley, and R. Wasserman, editors, *metametaphysics*, pages 347–383. Oxford University Press, April 2009.

[85] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[86] B. Schuller, S. Steidl, and A. Batliner. The interspeech 2009 emotion challenge. In *Proceedings of Interspeech*, pages 312–315, Brighton, UK, September 2009.

[87] L. She and J. Y. Chai. Interactive learning of grounded verb semantics towards human-robot communication. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1634–1644, Vancouver, Canada, July-August 2017.

[88] L. She, S. Yang, Y. Cheng, Y. Jia, J. Y. Chai, and N. Xi. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *Proceedings of the SIGDIAL 2014 Conference*, pages 89–97, Philadelphia, U.S.A., June 2014.

[89] J. M. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61:39–91, 1996.

[90] A. D. M. Smith and K. Smith. *Cross-Situational Learning*, pages 864–866. Springer US, Boston, MA, 2012. ISBN 978-1-4419-1428-6. doi: 10.1007/978-1-4419-1428-6_1712. URL https://doi.org/10.1007/978-1-4419-1428-6_1712.

[91] K. Smith, A. D. M. Smith, and R. A. Blythe. Cross-situational learning: An experimental study of word-learning mechanisms. *Cognitive Science*, 35(3):480–498, 2011.

[92] L. Smith and C. Yu. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106:1558–1568, 2008.

[93] M. Spranger. The co-evolution of basic spatial terms and categories. In L. Steels, editor, *Experiments in Cultural Language Evolution*, pages 111–141. John Benjamins, Amsterdam, 2012.

[94] M. Spranger. Grounded lexicon acquisition - case studies in spatial language. In *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL-Epirob)*, Osaka, Japan, 2013.

[95] M. Spranger. Incremental grounded language learning in robot-robot interactions - examples from spatial language. In *Proceedings of the 5th International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, Providence, USA, August 2015.

[96] L. Steels. Language games for autonomous robots. *IEEE Intelligent Systems*, 16(5): 16–22, September-October 2001.

[97] L. Steels and M. Loetzsch. The grounded naming game. In L. Steels, editor, *Experiments in Cultural Language Evolution*, pages 41–59. John Benjamins, Amsterdam, 2012.

[98] F. Stramandinoli, A. Cangelosi, and D. Marocco. Towards the grounding of abstract words: A neural network model for cognitive robots. In *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, July-August 2011.

[99] J. Strom, A. Richardson, and E. Olson. Graph-based segmentation for colored 3D laser point clouds. In *International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

[100] F. Stulp, E. A. Theodorou, and S. Schaal. Reinforcement learning with sequences of motion primitives for robust manipulation. *IEEE Transactions on Robotics (T-RO)*, 28(6):1360–1370, December 2012.

[101] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[102] A. Taniguchi, T. Taniguchi, and A. Cangelosi. Cross-situational learning with bayesian generative models for multimodal category and word learning in robots. *Frontiers in Neurorobotics*, 11, 2017.

[103] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4):64–76, 2011.

[104] S. Tellex, P. Thaker, R. Deits, D. Simeonov, T. Kollar, and N. Roy. Toward information theoretic human-robot dialog. In *Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.

[105] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori. An open-source simulator for cognitive robotics research: The prototype of the icub humanoid robot simulator. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pages 57–61, Gaithersburg, USA, August 2008. doi: 10.1145/1774674.1774684.

[106] Toyota Motor Corporation. *HSR Manual*, 2017.4.17 edition, April 2017.

[107] H. van Hasselt and M. A. Wiering. Reinforcement learning in continuous action spaces. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, Honolulu, USA, April 2007.

[108] P. Vogt. Exploring the robustness of cross-situational learning under zipfian distributions. *Cognitive Science*, 36(4):726–739, May 2012.

[109] C. Yu and D. H. Ballard. A unified model of early word learning: Integrating statistical and social cues. *Neurocomputing*, 70(13-15):2149–2165, August 2007.

[110] E. N. Zalta. Unifying three notions of concepts. *Theoria*, 87(1):13–30, February 2021.