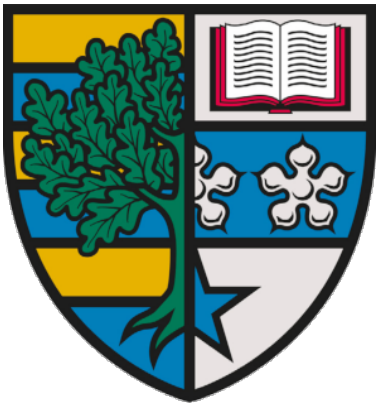
# Learning to Handle Miscommunication in Multi-Modal Conversational AI



Javier Chiyah-Garcia

Submitted for the degree of

*Doctor of Philosophy*

Department of Computer Science

School of Mathematics and Computer Sciences

Heriot-Watt University

April 2025

# Abstract

In human communication, we continuously negotiate shared understanding and deal with misunderstandings as they arise to achieve mutual coordination. However, despite the ubiquity and importance of misunderstandings and repairs in dialogue, conversational AI often struggles to process them effectively, limiting their ability to collaborate with humans through natural language.

This thesis explores how to develop robust models for processing miscommunications in situated collaborative tasks. We first collect a dialogue corpus to study human-agent coordination in an ambiguous environment, finding that models struggle to resolve referring expressions. To address this shortcoming, we design and train models to ground referring expressions and detect ambiguities, learning strong multi-modal representations in situated dialogues. We then analyse the signals required for models to learn to handle miscommunications, and propose a cross-modal taxonomy of clarifications to assess the contribution of distinct modalities. Our experiments with different model architectures and training objectives reveal that secondary objectives are essential to integrate multiple modalities (dialogue, visual and relational), leading to models better suited to deal with challenging clarifications in conversations. Finally, we evaluate how generative multi-modal LLMs handle both miscommunications and repairs by releasing a new benchmark, BlockWorld-Repairs, based on human data collections and studies. We then propose alternative training approaches that encourage models to learn from interactive settings, generalising to handling both instructions and subsequent repairs for successful task-completion. Throughout this thesis, we highlight the challenges posed by miscommunications and present approaches to develop robust collaborative conversational AI models better adapted for human interactions.

# Acknowledgements

I want to start by thanking Helen Hastie, who has offered invaluable support and guidance since we first met back in 2017. I would have never imagined that I would be where I am today without her encouragement, she saw my potential and pushed me to achieve it right from that first summer internship. I will be forever grateful and will never forget her kindness and the occasional push to achieve more. Despite not being part of the supervisory team in the end, she widely shaped the direction of this thesis, which simply would not have been possible without her.

I would like to thank my plethora of supervisors over the years, who have all played a role in shaping me into the person and researcher that I am today. First, Alessandro Suglia, who joined the supervisory team when I needed that extra help with modelling representations, but quickly became a pillar of the research. A huge thank you to him for his guidance and support, for being a constant source of inspiration and for answering my countless questions, often late at night or over weekends. I am grateful to have been able to work with and learn so much from him.

I would also like to thank my second supervisor, Arash Eshghi, for his guidance in linguistics and *grounding* this thesis. I will particularly cherish our uplifting conversations that always left me feeling motivated and inspired, even if they sometimes went on for a while! Every meeting with him motivated me to achieve more and think critically about the research, leaving no stone unturned.

To my other supervisors, José Lopes and Katrin Lohan, thank you both for providing different perspectives on the research and helping me navigate fields I was not familiar with. This thesis has a little bit of each of you in it, and I appreciate the role you both played. I would also like to thank my examiners, Simon and Yannis, who critically tested this thesis, but in particular to Yannis, whose feedback over the journey, albeit sometimes harsh, helped me iron out the finer details and shortcomings of my approach.

I would like to thank my industry funder, Siemens and particularly Georg von Wichert, as without their support, I would not have had the resources to pursue this research.

Next, I want to thank my family and friends for their support over the years. To my parents, thank you for your support throughout this journey and the years before, and to my brother for being a source of motivation. I am very fortunate to have the family I do, and I am grateful for their love and support. I am also grateful for my friends,

who have been there for me through thick and thin, particularly to Reuben, who cooked more than his fair share of meals alongside thought-provoking conversations; Birthe, who successfully distracted me from the PhD more than once (for my own good!); and Jules, who was always there for a chat when I needed it.

I am thankful to the people in my lab, who have been a constant source of inspiration through endless lunch (and meeting) discussions. Particularly those that I have been lucky enough to know over the years: Angus, Marios and Shubham; and those that I have worked with more recently, soon-to-be doctors themselves: Bruce, Amit, Nikolas, George, Malvina and Sabrina. You all played a part in shaping the ideas in this thesis through our critical conversations.

I am also grateful to the people I met and collaborated with during my internship at Amazon: Prasoon, Michael and Reza. And to the friends I made while in the US, especially Alexis, Miguel and Red, thank you for your friendship and support, particularly during those times I was probably working a bit too much.

There are also many others who have helped me in this journey and may not be mentioned here, but if you knew me at some point in the last few years, you probably played a role, and I am grateful for that.

Finally, I would like to thank Kieran[1], who has been there for me and supported all my decisions over the years. Despite ignoring my warnings and diving into a PhD himself, I hope he knows how fortunate I feel to have had him in this journey. My experience over these past few years would have been vastly different without him.

[1] My dear good 'friend'

# Table of Contents

# List of Tables

# List of Figures

# Glossary

**AI** Artificial Intelligence.

**AMTurk** Amazon Mechanical Turk.

**BCE** Binary Cross-Entropy.

**BW-R** BlockWorld-Repair.

**CE** Clarificational Exchange.

**CNN** Convolutional Neural Network.

**CR** Clarification Request.

**DPO** Direct Preference Optimization.

**FFN** Feed-Forward Network.

**GenAI** Generative AI.

**HCN** Hybrid-Code Networks.

**KL divergence** Kullback-Leibler divergence.

**LLM** Large Language Model.

**LM** Language Model.

**LoRA** Low-Rank Adaptation.

**LSTM** Long Short-Term Memory.

**MLP** Multi-Layer Perceptron.

**NLG** Natural Language Generation.

**NLI** Natural Language Inference.

**NLP** Natural Language Processing.

**NLU** Natural Language Understanding.

**POS** Part of Speech.

**PPO** Proximal Policy Optimisation.

**QLoRA** Quantisation Low-Rank Adaptation.

**REC** Referring Expression Comprehension.

**REG** Referring Expression Generation.

**RL** Reinforcement Learning.

**RLHF** Reinforcement Learning from Human Feedback.

**RNN** Recurrent Neural Network.

**RoI** Region of Interest.

**SFT** Supervised fine-tuning.

**SIMMC 2.0** Situated Interactive MultiModal Conversations 2.0.

**TPR** Third Position Repair.

**VLLM** Vision-Large Language Model.

**VLM** Vision-Language Model.

**VLN** Visual Language-Navigation.

**VQA** Visual Question Answering.

# List of Publications

Chiyah Garcia, Francisco Javier, José Lopes, and Helen Hastie (2020a). Natural Language Interaction to Facilitate Mental Models of Remote Robots. In: *Workshop on Mental Models of Robots*. HRI'20. Cambridge, UK: ACM.

Chiyah Garcia, Francisco Javier, José Lopes, Xingkun Liu, and Helen Hastie (2020b). CRWIZ: A Framework for Crowdsourcing Real-Time Wizard-of-Oz Dialogues. English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Marseille, France: European Language Resources Association, pp. 288–297.

Chiyah Garcia, Francisco Javier, Simón C. Smith, José Lopes, Subramanian Ramamoorthy, and Helen Hastie (2021). Self-Explainable Robots in Remote Environments. In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '21 Companion. Boulder, CO, USA: Association for Computing Machinery, pp. 662–664.

Chiyah-Garcia, Javier, Prasoon Goyal, Michael Johnston, and Reza Ghanadan (2024a). Adapting LLM Predictions in In-Context Learning with Data Priors. In: *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*. Ed. by Sachin Kumar, Vidhisha Balachandran, Chan Young Park, Weijia Shi, Shirley Anugrah Hayati, Yulia Tsvetkov, Noah Smith, Hannaneh Hajishirzi, Dongyeop Kang, and David Jurgens. Miami, Florida, USA: Association for Computational Linguistics, pp. 305–316.

Chiyah-Garcia, Javier, Alessandro Suglia, and Arash Eshghi (2024b). Repairs in a Block World: A New Benchmark for Handling User Corrections with Multi-Modal Language Models. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 11523–11542.

Chiyah-Garcia, Javier, Alessandro Suglia, Arash Eshghi, and Helen Hastie (2023). 'What are you referring to?' Evaluating the Ability of Multi-Modal Dialogue Models to Process Clarificational Exchanges. In: *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani. Prague, Czechia: Association for Computational Linguistics, pp. 175–182.

Chiyah-Garcia, Javier, Alessandro Suglia, José David Lopes, Arash Eshghi, and Helen Hastie (2022). Exploring Multi-Modal Representations for Ambiguity Detection & Coreference Resolution in the SIMMC 2.0 Challenge. In: *AAAI 2022 DSTC10 Workshop*.

Lopes, José, Francisco Javier Chiyah Garcia, and Helen Hastie (2020). The Lab vs The Crowd: An Investigation into Data Quality for Neural Dialogue Models. In: *Workshop on Human in the Loop Dialogue Systems at NeurIPS 2020*.

Wilson, Bruce W, Shivaanee Eswaran, Omar Riyaz, Javier Chiyah-Garcia, and Matthew Peter Aylett (2024). Follow the Yellow or Red Brick Road? Investigating the Impact of Narratives in a Guided Navigation Task. In: *Proceedings of the 12th International Conference on Human-Agent Interaction*. HAI '24. Swansea, United Kingdom: Association for Computing Machinery, pp. 411–413.

# Chapter 1

# Introduction

Conversational agents have gained significant popularity nowadays, promising the potential to achieve tasks quickly and effortlessly through speech. Natural language could be seen as the clearest and most efficient interface for engaging with machines, as it mirrors human communication and eliminates the need for specialised knowledge to navigate unfamiliar systems. In practice, however, these systems are often brittle and limited, requiring users to simplify their language (Wu et al., 2020b), or are prone to failures (Wright et al., 2020; Nesset et al., 2021), acting on uncertainty rather than clarifying, as a human would. Therefore, the goal of interacting with agents as naturally as with other people remains largely unrealised, relegating conversational systems to basic or non-interactive tasks, such as checking the weather or summarising text.

Human communication relies heavily on establishing and maintaining *common ground*, the set of shared contexts, experiences and knowledge crucial for mutual understanding (Clark, 1996; Clark and Brennan, 1991; Goodwin, 1981). However, language only provides a limited window into our thought processes, often abstracting away many details from what we say and leaving much open to interpretation. We are indeed particularly good at inferring information from the conversational context (Grice, 1969), drawing on surroundings, personal experiences and even assumptions about others' intentions to attribute meaning to an utterance (Bender and Koller, 2020). Producing underspecified utterances (e.g., *"Bring a mug"*) is cognitively less demanding than fully specified ones (e.g., *"Bring a clean mug from the kitchen to serve you coffee"*) (Levinson, 2000), hence language is often underspecified (Pezzelle, 2023).

Consider the dialogue example below:

(1) ***Person:*** *Pick up the small block.*

***Agent:*** *Which one? There's a blue and a gold one.*

***Person:*** *The one on the right.*

***Agent:*** [moves towards a cylinder].

***Person:*** *No, not that one! The cube!*

***Agent:*** [picks up the small blue cube on the right].

This underspecification makes natural language communication a highly collaborative process between interlocutors, who have to signal sufficient understanding to build the common ground and progress in the conversation. Thus, there is a continuous exchange of positive and negative feedback, called *communicative grounding* (Clark, 1996), to coordinate actions within dialogues (Mills, 2014; Eshghi et al., 2022). However, conversations are inherently spontaneous, leading speakers to occasionally produce malformed or incorrect statements that shift the intended meaning of an utterance. When participants become aware of a communicative breakdown, or miscommunication, they quickly trigger corrective mechanisms to resolve misunderstandings (Dingemanse et al., 2015; Enfield et al., 2013). These *repairs* (Schegloff et al., 1977; Schegloff, 1992) help clarify miscommunications and realign communicators' perspectives, making them essential for achieving mutual coordination (Mills, 2007) and effective communication.

However, despite the inherent collaborative nature of dialogue and the frequency of miscommunication (Colman and Healey, 2011), existing conversational agents struggle to interpret these phenomena in human interactions, with few works even considering the interactivity of dialogues (Schlangen, 2023). This is particularly problematic if we aim to collaborate with machines as highly capable agents across diverse settings (Suglia et al., 2024), such as embodied human-robot tasks, where the conversational context may include objects or actions in our surroundings. In these embodied tasks, it is critical to establish joint attention to objects or events in the environment (Tomasello, 1995) to successfully refer to them and coordinate actions. Here, the context is significantly visual, with perceived objects, their properties (like colour, shape or location), and actions forming a large part of the common ground. While the linguistic context (instructions, descriptions or feedback) directs attention and specifies goals, it relies heavily on this

shared visual perception for disambiguation. In such cases, an agent not only has to deal with communicative grounding within the dialogue but also *symbol grounding* (Harnad, 1990), connecting language (symbols) to their perception of the world (objects), often mediated by this established joint attention. Thus, situated dialogue becomes a complex, multifaceted process linking language, participants and the relevant aspects of the shared environment.

To fully grasp the conversational context, particularly in situated interactions where the visual context often provides crucial disambiguating information that the linguistic context alone cannot supply, agents must go beyond text alone (Bender and Koller, 2020) and learn to ground concepts or reason about the impact of their actions in the environment (Bisk et al., 2020). It is only by intertwining both symbol and communicative groundings (Larsson, 2018) that we can distil agents with truly collaborative capabilities. Therefore, the ability to interpret and generate effective repair sequences during dialogue is essential for cooperative conversational agents (communicative grounding), along with the capacity to link language references to perceived objects in the environment (symbol grounding).

In this thesis, we investigate dialogue miscommunications and their resolution within multi-modal settings for collaborative conversational agents. Specifically, we focus on referential miscommunications in human-agent interactions to understand what models need to resolve them, considering the relation between visual perception and language cues. We identify the relevant signals necessary for learning robust object representations, internal object models with disentangled visual attributes (colour, shape, type) that remain stable despite limitations in vision or language (e.g., partial occlusion, language underspecification). These representations enable models to address misunderstandings across both vision and language modalities, supported by a new taxonomy for analysing the contribution of modalities in repairing dialogues. Furthermore, we collect dialogues in a highly ambiguous environment and introduce BlockWorld-Repairs, a benchmark and dataset of situated dialogues designed to evaluate models' ability to process repairs. We demonstrate how custom training regimes lead to models that can generalise and handle both instructions and repairs. The findings presented here provide valuable strategies for developing conversational agents robust to multi-modal miscommunications in situated dialogues, paving the way towards true human-agent dialogue cooperation built on shared understanding and effective communication.

## 1.1 Research Questions

As part of this thesis, we aim to understand how conversational agents can learn to overcome the challenges posed by miscommunications. These particular dialogue phenomena impose constraints on the agents' usefulness in collaborative tasks, and hence we argue that learning to resolve multi-modal miscommunications is crucial for true human-agent collaboration. The main research question addressed in this thesis is:

**RQ:** ***How can models process referential miscommunications in multi-modal collaborative tasks?***

Resolving miscommunications involves integrating many interconnected aspects such as perception, reasoning and communication. Thus, we break down this central question into smaller ones, each addressing separate challenges explored throughout this thesis. We illustrate how these research questions are interconnected in Figure 1.1.

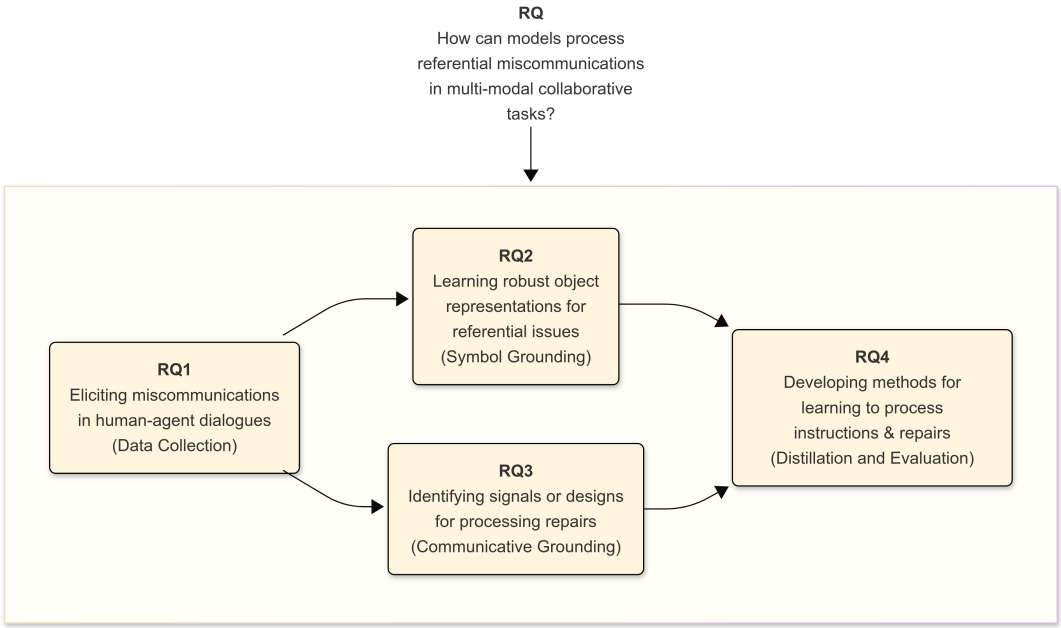



Figure 1.1: Visual breakdown of the research questions addressed in this thesis.

**RQ1:** ***What are the ways that we can elicit miscommunications in multi-modal referential conversations?*** As discussed, communication is a highly collaborative process, yet many existing works assume that agents function as passive instruction followers. It is, however, difficult to isolate factors impacting human-agent interactions in collaborative scenarios (i.e., *Can the agent ground the conversation? What about specific objects?*).

Only recently have pre-trained models become good enough to be used across diverse tasks and settings, enabling the exploration of genuinely interactive scenarios. Hence, we investigate a collaborative task that cannot be solved by better models alone, requiring the grounding of previous actions, objects and dialogue together (Chapter 3).

**RQ2:** ***How can models learn robust multi-modal representations to resolve ambiguous referring expressions?*** Agents must comprehend their surrounding environment to accurately identify and include objects into the conversational context. Connecting referenced objects (or actions) to their corresponding word in dialogue is complex, requiring agents to address the symbol grounding problem. It requires learning *robust object representations* capable of overcoming the limitations of the visual modality, such as disentangling distinct visual attributes (e.g., colour, shape), remain invariant to shifts in appearance or viewpoint (including partial occlusion) and capture discriminative features to distinguish between similar objects. This work will explore model representations to resolve references in visual dialogues, including the critical task of detecting ambiguities in such cases to trigger repairs (Chapter 4).

**RQ3:** ***What are the signals necessary for learning to process repairs?*** Miscommunications and their subsequent repairs offer insights into models' internal processes, as repairs involve precise revisions to prior predictions. We will study how different model architectures or training objectives influence models' ability to process repairs, probing the types of information that they leverage to resolve these issues in multi-modal scenarios (Chapter 5).

**RQ4:** ***How can we distil in models the capability to process repairs?*** Humans can seamlessly handle dialogues involving both instructions and repairs, yet models often struggle to generalise when these elements are combined. Furthermore, it remains unclear how models can effectively learn from cooperative dialogues, as not all strategies will be equally successful. This thesis will explore this final question by proposing a benchmark and training approaches aimed at improving models' generalisation capabilities, ultimately achieving better alignment with humans (Chapter 6).

## 1.2 Contributions

This thesis makes several contributions to understanding and modelling referential miscommunications in multi-modal collaborative tasks. First, we collected a novel dialogue corpus within the Blocks World environment, designed to elicit context-dependent corrections and highlight the challenges of resolving referring expressions in interactive settings. Second, using the SIMMC 2.0 dataset, we investigated methods for learning robust multi-modal representations to resolve ambiguous referring expressions, finding that visual information alone is often insufficient in cluttered scenes. Third, we analysed Clarificational Exchanges in multi-modal dialogues, proposed a fine-grained taxonomy for them and identified that object-centric representations improve models' ability to process referential repairs. Fourth, we introduced the BlockWorld-Repairs benchmark to assess VLMs' capabilities in handling complex instructions and repairs, showing that alternative fine-tuning strategies focusing on critical information can improve their performance when processing both instructions and repairs. Overall, our findings emphasise the need for models to learn from interaction, learn robust disentangled object representations and attend to relevant signals to effectively learn to process miscommunications in collaborative dialogues.

### *1.2.1 Ethical Concerns*

This thesis addresses human-agent interaction, opening up new possibilities while also introducing potential risks to consider. Our research involves simulated robotic systems, such as a robot arm collaborating on pick&place tasks. Even though these simulations simplify real-world robot movement to focus on specific actions (abstracting away the full range of motion), safety remains a key concern, especially when agents operate in shared spaces. We believe that it is ethically important for agents to handle miscommunications as it allows them to correct errors that pop up during collaborative tasks.

In Chapter 3, we built safety considerations in from the start, using techniques like Hybrid Code Networks (Williams et al., 2017), which are well-suited for human-robot interaction. We also collected dialogue data for this chapter using Amazon Mechanical Turk (AMTurk), simulating dialogue interactions. We acknowledge that despite careful design, data collected via crowdsourcing platforms may contain inherent biases reflect-

ing the worker pool or task framing. Similarly, the data from our in-person human study conducted in Chapter 6, designed to gather human performance data on these initial dialogues, may also reflect participant biases.

Beyond the data collection, the development and deployment of the models themselves raise ethical points. Models trained on human interaction data risk inheriting and even amplifying existing societal biases from that data. Additionally, understanding how these models interpret instructions and fix mistakes is crucial for building user trust and holding them accountable when things go wrong. While this thesis concentrates on making collaboration more robust by improving how agents handle misunderstandings, developing these systems responsibly means continually paying attention to these broader ethical issues.

## 1.3 Thesis Publications

This thesis resulted in several peer-reviewed research outputs published at international venues. The ones that are part of this thesis are the following:

1. Chiyah Garcia, Francisco Javier, et al. (2020b). CRWIZ: A Framework for Crowdsourcing Real-Time Wizard-of-Oz Dialogues. eng. *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

2. Chiyah-Garcia, Javier, et al. (2022). Exploring Multi-Modal Representations for Ambiguity Detection & Coreference Resolution in the SIMMC 2.0 Challenge. *AAAI 2022 DSTC10 Workshop*.

3. Chiyah-Garcia, Javier, et al. (2023). 'What are you referring to?' Evaluating the Ability of Multi-Modal Dialogue Models to Process Clarificational Exchanges. *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

4. Chiyah-Garcia, Javier, et al. (2024b). Repairs in a Block World: A New Benchmark for Handling User Corrections with Multi-Modal Language Models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

The research also fostered collaborations leading to additional peer-reviewed publications referenced throughout this document but beyond the scope of this thesis:

1. Chiyah Garcia, Francisco Javier, et al. (2020a). Natural Language Interaction to Facilitate Mental Models of Remote Robots. *Workshop on Mental Models of Robots*. HRI'20.

2. Lopes, José, et al. (2020). The Lab vs The Crowd: An Investigation into Data Quality for Neural Dialogue Models. *Workshop on Human in the Loop Dialogue Systems at NeurIPS 2020*.

3. Chiyah Garcia, Francisco Javier, et al. (2021). Self-Explainable Robots in Remote Environments. *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '21 Companion.

4. Chiyah-Garcia, Javier, et al. (2024a). Adapting LLM Predictions in In-Context Learning with Data Priors. *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*.

5. Wilson, Bruce W, et al. (2024). Follow the Yellow or Red Brick Road? Investigating the Impact of Narratives in a Guided Navigation Task. *Proceedings of the 12th International Conference on Human-Agent Interaction*. HAI '24.

## 1.4 Thesis Overview

We begin with Chapter 2 introducing essential background literature to support the rest of the thesis. We discuss the importance of communicative and symbol grounding, and provide further discussion on the role of miscommunications and repairs for cooperative communication. Additionally, we review the state of the art in language models, including transformers, pre-trained large language models and various training approaches. We also survey prior works on instruction understanding in situated dialogues, identifying the research gap this thesis addresses, and introduce research areas closely related to the processing of miscommunications in interactive dialogue, such as language games and embodied interaction.

In Chapter 3, we present one of the main environments used to explore our research questions and describe the collection of a corpus of collaborative human-agent dialogues rich in repairs for a challenging task. Chapter 4 then focuses on learning to resolve referring expressions in multi-modal dialogues, where we develop a model to deal with the previously discussed symbol grounding problem. We also train models to detect referential ambiguities preceding clarifications, a first step in handling miscommunications in dialogues. To understand what models require to resolve these miscommunications, Chapter 5 analyses the clarificational exchanges across different model architectures and training objectives. We propose a framework and taxonomy of clarifications to examine the fine-grained properties models exploit when resolving ambiguities, finding that certain architectures and objectives are more effective for resolving multi-modal clarifications.

Building on these insights, Chapter 6 introduces a novel benchmark to assess models' ability to process repairs. With this benchmark, we aim to develop models capable of handling miscommunications in collaborative dialogues, first evaluating their generalisation capabilities and later improving them with specialised training regimes that learn from relevant information. To benchmark models against human performance, we conduct a human study where participants assume the agent's role from Chapter 3, identifying areas where agents must improve to become better collaborators. Our research then concludes in Section 6.6 with additional experiments on how we can further distil generalisation capabilities with reinforcement learning with human feedback in conversational agents. Finally, Chapter 7 presents a summary of the main thesis findings and a discussion of future directions for research in this field.

# Chapter 2

# Background

This chapter provides an overview of the literature supporting this thesis, including the core problems addressed and situating our research within the broader research community. To investigate miscommunications with conversational agents, we begin by introducing the foundational concepts that systems using natural language communication have to deal with in Section 2.1, such as the communicative and symbol grounding problems. We discuss in detail the linguistic theories that support human communication and their relevance to artificial systems, particularly with regard to miscommunications and repairs as key pillars of cooperative human communication. As we use Deep Learning extensively throughout this work to design and train machine learning models, we summarise major findings and trends in the field in Section 2.2, tracing developments from neural networks to large language models.

Section 2.3 then presents the primary technical background for this thesis: following instructions in natural language. We review prior research on developing agents capable of following instructions and its connection to miscommunications and human language. Additionally, we discuss language games and embodied human-robot interaction, two areas where cooperation (and thus managing miscommunications) is crucial yet often oversimplified in current benchmarks, limiting their real-world transferability.

Many works have previously attempted to deal with the challenges associated with collaborative tasks; thus, in Section 2.3.4, we provide a comprehensive overview of relevant collaborative tasks, environment and datasets, comparing these to our work in this thesis. Finally, we conclude this chapter in Section 2.4, summarising the reviewed literature and outlining how it informs our approach to the research questions of Section 1.1.

## 2.1 Communicative Grounding

Human conversations are an interactive process between speakers and addresses (Schlangen, 2023), who continuously work together to coordinate mutual understanding (Clark and Schaefer, 1989; Goodwin, 1981; Clark, 1996). Dialogues are supported by a broad range of assumptions that enable effective communication, that is the collection of shared knowledge, experiences, beliefs, and even goals of the interlocutors. We estimate these unconsciously, but they attribute meaning beyond words alone which is crucial to understanding the interaction. This is what Stalnaker (1978) refers to as *common ground* or the accumulated context of the conversation:

> 'The propositions whose truth he [the speaker] takes for granted as part of the background of the conversation'

During a dialogue, speakers contribute utterances that iteratively build the common ground, expanding the dialogue context with a set of propositions mutually taken for granted by participants (Stalnaker, 1978). Subsequent utterances, as long as they do not provide negative feedback (i.e., contradict prior statements), implicitly assume that the information from previous contributions has been accepted into the common ground. Stalnaker (2002) further elaborates on this and frames common ground from the speaker's perspective: it comprises what the speaker presupposes is mutually believed as an informational state. Assertions then serve to update this presupposed shared state, highlighting its reliance on individual belief about mutual acceptance.

In contrast to viewing common ground primarily as a shared informational state, Clark (1996) instead proposes *grounding* as the underlying collaborative process:

> '[Grounding] is a sine qua non for everything we do with others – from the broadest joint activities to the smallest joint actions that comprise them.'

Thus, from this perspective, grounding is not merely about the static set of shared beliefs but the dynamic, moment-by-moment process through which interlocutors actively work to reach mutual understanding. Interlocutors provide evidence of their understanding, seek confirmation and collaboratively add information to what they believe is mutually understood. In this view, participants provide evidence (Clark and Brennan, 1991) that leads them to believe the information can be added to the common ground.

This makes dialogue interactions an inherently collaborative process (Clark and Schaefer, 1989), where participants have to actively engage by not only interpreting what is said but also producing timely and relevant utterances. Therefore, participants coordinate through a continuous exchange of both positive and negative feedback, to establish what has been sufficiently understood and can now be part of the common ground. Clark (1996) and related works (Clark and Brennan, 1991; Clark and Schaefer, 1989) describe this joint process as grounding, which we henceforth refer to as *communicative grounding*. It entails interlocutors perceiving what is said, comprehending their context-dependent meaning and finally collaboratively agreeing on their addition to the shared understanding, often signalled implicitly by moving to the next relevant contribution (i.e., producing a new utterance that expands upon the previous one).

Achieving mutual understanding often relies on establishing *joint attention* of all the interlocutors towards relevant aspects of the context or topic (Clark, 1996; Tomasello, 1995). Thus, the shared environment provides crucial resources for grounding (Clark and Marshall, 1981) as the backdrop against which common ground is collaboratively built.

Consider the dialogue example below:

(2) ***Person A:*** *We have dinner booked for seven, bring an umbrella.*

***Person B:*** *I can pick you up in the car at six.*

***Person A:*** *It will be busy, so it's better to walk.*

***Person B:*** *Umbrella and boots it is!*

Initially, Person A implicitly proposes to walk by mentioning the umbrella, to which Person B replies with an alternative plan without acknowledging it, instead saying that they can pick up A in the car. Person A then produces an utterance stating that it will be busy and that walking is likely a better option, with Person B finally agreeing to A's perspective. By the end of this short conversation, Person A and B have both grounded the following: 1) they are meeting for dinner possibly that same day; 2) it will be raining; 3) the place will be busy; 4) Person B is willing to take the car; and so on... This excerpt shows that both interlocutors were actively participating in the dialogue, iteratively grounding new information without producing utterances with direct positive or negative feedback (e.g., *"I understood!"*, *"Yes"* or *"I disagree"*).

Clark and Schaefer (1989) propose that discourse contributions occur when inter-

locutors arrive at a mutually satisfactory interpretation of an utterance. They divide the contribution process in two: a *presentation* and an *acceptance* phase. A speaker *presents* an utterance that the addressee accepts by providing evidence of understanding. In the case above, Person A first presents an utterance (*"We have dinner booked for seven, bring an umbrella"*), which Person B accepts with utterance (*"I can pick you up in the car at six"*).

Interlocutors may provide two main types of evidence of mutual understanding, positive and negative (Clark and Brennan, 1991; Clark and Schaefer, 1989), across different communication channels. For instance, by producing an utterance that evidences understanding that proceeds to the next contribution (e.g., *"I can pick you up in the car at six"*); explicit acknowledgements (e.g., *"Yes"*/*"No"*); or backchannels (e.g., nods, eye gaze). Negative feedback, such as asking for clarification (e.g., *"Did you say dinner at seven?"*) or other backchannels (e.g., *"huh?"*), indicates that the participant cannot *accept* or agree to what has been said and requires further negotiation before it can be grounded in the conversation.

### 2.1.1 Understanding Issues and Clark's Action Ladder

Negative feedback during the acceptance phase signals communication issues that interlocutors must address to advance the discourse. At this stage, one of the interlocutors provides evidence of the issue, which the conversation partner can use to formulate a new utterance aimed at resolving it. This process is recursive but not infinite, as each subsequent iteration requires progressively weaker evidence, adhering to Clark's 'Strength of evidence principle' (Clark and Schaefer, 1989).

Both Clark (1996) and Allwood (1995) independently proposed classifications for the levels of understanding in communication. This hierarchy of actions, referred to as the *joint action ladder*[1] (see Table 2.1), represents successively stronger states of evidence, where each level builds on the acceptance of the previous one to advance to the next:

Each level in this ladder is useful to determine what obstacles prevent an utterance from being grounded in the conversation. The classification itself is also grounded in modalities as to include both linguistic and non-linguistic actions. For instance, lack

[1]Throughout this thesis, we will use Clark's (1996) definition of the joint action ladder, while also incorporating the insights of Allwood (1995).

| **Level** | Clark (1996) | Allwood (1995) | **Description** |
|---|---|---|---|
| 1 | Execution and attention | Contact | Addressee did not notice the utterance (e.g., *"Are you talking to me?"*) |
| 2 | Presentation and identification | Perception | Addressee partially misses the utterance (e.g., *"What did you say?"*) |
| 3 | Meaning and understanding | Understanding | Addressee does not understand part of the utterance (e.g., *"The green or yellow shirt?"*) |
| 4 | Proposal and consideration | Reaction | Addressee misunderstands intentions (e.g., *"Are we leaving now?"*) |

Table 2.1: Clark (1996) and Allwood (1995) joint action ladder for the levels of communication.

of attention signals issues with Level 1, that of execution and attention. Benotti and Blackburn (2021) describes the modalities as:

**Level 1:** Socioperception, the process of perceiving and interpreting social cues (Tomasello et al., 2005).

**Level 2:** Hearing or speech recognition.

**Level 3:** Vision for recognising real-world referents.

**Level 4:** Kinesthetic for moving or acting.

This thesis centres on understanding and addressing challenges at Level 3 of this joint action ladder. Our investigation examines dialogue coordination and the obstacles passed by referential ambiguities, which arise from underspecified references in real-world contexts. Referring in conversations is itself a collaborative process as highlighted by Clark and Wilkes-Gibbs (1986).

While communicative grounding focuses on interactively establishing shared understanding in dialogue between interlocutors, a related but distinct challenge involves connecting the linguistic symbols used in that dialogue to the perceptual world, the symbol grounding problem. The next subsection explores the role of vision in these challenges, how vision and language interplay and its connection to our work.

### 2.1.2 Symbol Grounding

A different, but closely related problem, is that of perception. The ability to perceive the world and relate it to our internal language representation is what Harnad (1990) defined as the *symbol grounding problem*:

> 'How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?'

Harnad (1990) identified[2] that any artificial agent that used language to communicate would have to connect the fixed, high-level features of physical objects (a concept or meaning) to their corresponding symbolic representation (word). Associating the meaning of a word with its concept is the result of an agent's experience with it. For instance, a person who has seen a written word (e.g., a flower) and a real-world instance of that object (a plant with petals) will not be able to associate them together until these terms are grounded. Importantly, learning the perceptual meaning of a word would allow an agent to *extend* it to other entities, i.e., a flower and the set of all flowers. This enables agents (and humans) to identify new instances of that object without necessarily perceiving all possible combinations (e.g., red vs blue flower).

Symbol grounding fundamentally deals with linking abstract symbols to perceptual experiences or sensorimotor representations of the world. According to Barsalou (1999), the grounding mechanism itself may be rooted in the perceptual and motor states experienced during interaction with the world, directly linking language and concepts to sensorimotor systems. Thus, there is a strong connection between grounding, embodiment and cognition (Barsalou, 2008).

Solving the symbol grounding problem implies mapping words to what is seen, heard or felt, and thus requires agents to deal with three tasks (Harnad, 1990): learn concepts, relate them to symbols and combine them to create other concepts, or *compositional semantics* (Montague, 1970). We can derive new concepts from previous ones, such that a concept like 'left' or 'right' can be combined with 'top' to obtain 'top-left', inheriting the meaning of both. Humans leverage compositionality to continuously learn novel

[2]Following the earlier Chinese room theory by Searle (1980).

concepts based on simpler ones, transferring context across experiences. This flexibility is essential for general agents to be grounded in the world.

Both symbol grounding and communicative grounding are closely related (Larsson, 2018), with the first dealing with the dynamics of the environment (perception, relation) and the latter with the dynamics of dialogue (dialogue interaction, pointing). Importantly, (Larsson, 2018) argues that symbol grounding can be a direct result of communicative grounding within a mutual environment, and thus communicative grounding facilitates symbol grounding (e.g., learning the meaning of 'wug' by linking it to the perceived object). It is the interaction process that enables the perceptual connection between language and the world, cutting across modalities (Benotti and Blackburn, 2021) (i.e., communicative grounding does not exclusively use language, but may involve gestures, such as pointing or nodding, for coordination or acknowledgement).

Consider this example:

(3) ***Person A:*** *Can you pass the wug?*

***Person B:*** *Huh?*

***Person A:*** *[pointing at an object] That on the table*

In this case, Person B learns that 'wug' refers to a particular object that Person A pointed to, grounding the meaning (physical object) to the symbol (word 'wug'), as a result of communicative grounding. Linguistic cooperation in this case led to learning a novel perceptual meaning.

Symbol grounding is particularly relevant in situated interactions, where language refers to objects or features of the immediate perceptual environment. As mentioned earlier, communicative grounding requires establishing joint attention of all the interlocutors towards the relevant context. Similarly here, joint attention is the mechanism through which specific referents are singled out for grounding (e.g., pointing at the 'wug' object above).

Conversational agents must therefore manage both types of grounding when interacting in shared visual environments with humans, acknowledging that grounding arises from interaction and perception (Steels, 2008). The work in this thesis deals with their intersection when issues arise regarding objects in the environment. Thus, we focus on the interplay of symbol grounding and communicative grounding during communicative

breakdowns. We however do not address the learning of concepts or compositionality; for further discussion on compositionality refer to Suglia et al. (2024) instead.

### 2.1.3 Miscommunications

Conversations are highly coordinated activities in which participants collaborate to achieve a shared goal or joint action (Clark and Schaefer, 1989). However, by their inherent nature, conversations are susceptible to a wide range of challenges and obstacles. Speakers produce utterances spontaneously in ever-changing environments, typically without a predefined plan, while addressees must interpret these utterances in real time to collaborate effectively.

Interlocutors rely on shared contexts, such as experiences, knowledge and goals, to establish mutual understanding. Humans excel at inferring information from the overall conversational context (Grice, 1969), and thus we prefer to produce utterances that are easier to produce at the cost of harder to understand. Shorter, underspecified utterances are cognitively less demanding than fully specified ones (Levinson, 2000). Consider this example:

(4) ***(i)*** *Bring a mug.*

***(ii)*** *Bring a mug from there.*

***(iii)*** *Bring a clean mug from the kitchen to serve you coffee.*

Utterance (iii) provides significantly more details than (i) and (ii), but depending on the context, all three may be equally effective in identifying the referent (a mug). However, producing utterance (iii) incurs a considerable time and cognitive effort for the speaker, which may prefer minimising effort despite the difficulties it may cause for the listener to understand the utterance. This phenomenon is referred to as *semantic underspecification* (Ferreira, 2008; Frisson, 2009) and occurs when when the full meaning of an utterance (or linguistic structure) relies on contextual cues to resolve its ambiguity. Language is inherently underspecified (Pezzelle, 2023), which improves communication efficiency (Piantadosi et al., 2012) by preventing speakers from articulating every detail explicitly.

These ambiguities or other misunderstandings (e.g., hearing difficulties) however can lead to breakdowns in communication, or *miscommunications*, preventing further com-

municative grounding. Miscommunications arise in conversation when one of the interlocutors detects a problem in their own or the other's understanding and signals an understanding issue. Miscommunications can occur at any level of Clark's (1996) joint action ladder, and interlocutors address them with highly structured corrective feedback mechanisms: repairs (Schegloff et al., 1977; Schegloff, 1992).

### *2.1.4 Repairs*

Repairs play a crucial role in resolving misunderstandings (Enfield et al., 2013) and realigning communicators' perspectives, making them essential for achieving mutual coordination (Mills, 2007) and effective communication. However, repairs are highly complex and context-dependent, often allowing for multiple interpretations (Purver, 2004; Purver and Ginzburg, 2004) and frequently involving multiple modalities (Benotti and Blackburn, 2021). They are also a common feature of human dialogue, occurring approximately 1.4 times per minute on average (Dingemanse et al., 2015), with their frequency influenced by the nature of the task and the communication medium (Colman and Healey, 2011). We provide an example repair in example 5.

(5) ***Person A:*** *Can you bring my jacket from the living room?*

***Person B:*** *The red jacket?*

***Person A:*** *No, the blue one.*

Repairs are particularly significant in language modelling because they almost always involve highly focused and specific revisions to the representations already built by a model or agent, based on the context-dependent structure of the corrective feedback. For instance, in dialogue 5, Person B initially inferred that Person A was referring to the red jacket. However, A clarified that they were referring to a different one, requiring Person B to revise their expectations. Prior works have used repairs as probes into how models represent meaning and the structure of the overarching task (e.g., Madureira and Schlangen, 2023; Benotti and Blackburn, 2021), investigating how different modalities influence these interaction. Similarly, in this work, we explore these aspects in Chapter 5 and Chapter 6.

Depending on who initiates or triggers the repair, and who resolves it, the literature categories repairs in a taxonomy outlined in Table 2.2. The terms 'self' and 'other' refer

to the speaker or the addressee of the problematic utterance, respectively. For example, self-initiated self-repairs occur when the speaker both triggers the repair and solves the issue, sometimes within the same conversational turn, see example in Table 2.2. For a more extensive discussion on types of repairs, refer to Albert and Ruiter (2018).

| **Initiated** | **Completed** | **Example** |
|---|---|---|
| Self-initiated | Self-repair | A: I want to book a trip to Paris, eh, Rome |
| Self-initiated | Other-repair | A: I remember going to ehh, somewhere near Madrid, a small city<br>B: Toledo?<br>A: That's it |
| Other-initiated | Self-repair | A: Did you ever go to Edinburgh whilst in Scotland?<br>B: Huh?<br>A: Did you go to Edinburgh? |
| Other-initiated | Other-repair | A: If we are going to be late we'll have to buy fish cos we won't have time to cook anything<br>B: Oh fish and chips you mean... from the chippie<br>B: Yeah |

Table 2.2: Taxonomy of repairs with some examples from Schegloff et al. (1977).

Some repair types are more prevalent than others in dialogue, with self-repairs being the most common while other-initiated other-repairs the least common (Schegloff et al., 1977). Given the significant structural variability across repair types, this work focuses on two key categories: *Self-Initiated Self-Repairs* and *Other-Initiated Self-Repairs*. In both of these cases, we specifically examine repairs that are not within the initial conversational turn, so the communication breakdown is evident only after the addressee has had a chance to produce an utterance (i.e., Speaker $\rightarrow$ Addressee $\rightarrow$ Speaker).

This work specifically focuses on repairs occurring in the third position (P3) of the repair sequence, following the framework by Healey et al. (2005). P3 repairs are initiated after the initial speaker has completed their turn and the addressee has responded (i.e., Speaker $\rightarrow$ Addressee $\rightarrow$ Speaker). P3 repairs are highly significant in interactions: unlike repairs initiated within the first turn (e.g., *"I want to book a trip to Edinburgh, eh, Glasgow"*) or second turn (P2)[3], P3 repairs signal communication breakdowns that only become evident after an explicit opportunity for the addressee to signal understanding or

[3]P2 repairs are often followed by a follow-up in P3 (Healey et al., 2005), e.g., *"past a forge on your right? – past a what?"*.

indicate a problem. Analysing these sequences provides insights into how participants manage more complex grounding issues that persist across turns.

Within P3 repairs, we further narrow our focus to Self-Initiated Self-Repairs (P3SISR) and Other-Initiated Self-Repairs (P3OISR). Both P3SISR and P3OISR frequently occur in task-oriented dialogues (Colman and Healey, 2011), particularly instruction-following tasks such as the HCRC Map Task Corpus (Anderson et al., 1991)[4]. Thus, P3 lie within our goal to investigate miscommunications in collaborative, task-focused dialogue interactions with multi-modal models.

### Self-Initiated Self-Repairs

These self-repairs are self-initiated by the same speaker that produced the problematic utterance, and thus are highly frequent in dialogues. We focus on a particular type in this thesis, **Third Position Repair (TPR)**, which occur on the third turn in a dialogue after the misunderstanding is clear to the original speaker. Consider the following example:

*(6)* ***Person A:*** *Can you pick me up after work to go shopping?*

***Person B:*** *Sure, I'll come by the house around 5 pm.*

***Person A:*** *Oh, not there. I'm seeing a friend, so pick me up at the train station instead.*

Person A is not aware of the mismatch of expectations until Person B produces an utterance with evidence of the task to solve. More common are referential ambiguities such as:

*(7)* ***Person C:*** *What's all this?*

***Person D:*** *mmh, something I picked up.*

***Person C:*** *No, I meant that.*

In this highly underspecified dialogue, Person D reveals a referential ambiguity by referring to an unexpected object for Person C, who subsequently resolves it with a TPR, pointing at something else. In Chapter 3 and Chapter 6, we explore computational models for processing these types of repairs.

[4]We will discuss these in more detail in Section 2.3.4.

### Other-Initiated Self-Repairs

Unlike the previous repairs, these are other-initiated, meaning the addressee identifies and raises the misunderstanding when they cannot fully understand or accept the utterance produced by the initial speaker, which the speaker corrects in the third turn.

In this thesis, we focus on a specific type of repair raised with **Clarification Request (CR)**, where the addressee produces a request for clarification (CR) and the original speaker responds with additional information or a correction in the subsequent turn. Consider the example below:

(8) ***Person A:*** *Can you bring my jacket from the living room?*

***Person B:*** *Which one?*

***Person A:*** *The red jacket.*

Person A produced an utterance that was ambiguous or too underspecified for Person B to interpret, who subsequently asks a CR (*"Which one?"*) prompting Person A to provide additional details (*"The red jacket"*).

We define this brief meta-communicative exchange between interlocutors as a **Clarificational Exchange (CE)**: an initial utterance, followed by a CR and a subsequent clarification. In this work, we examine such repairs and clarifications in Chapter 4 and Chapter 5 for resolving ambiguous referring expressions.

In Chapter 5, we present a taxonomy of CRs based on the modalities used to resolve them. While Clark's (1996) joint action ladder provides a foundation of communication, it lacks the granularity needed to analyse specific impact of each modality on a model's ability to resolve miscommunications. Similarly, Schlangen (2004) proposed a theoretical model categorising processing issues leading to CRs, as outlined in Table 2.3. Although this classification effectively distinguishes fine-grained problems that trigger CRs, it does not consider the modalities employed to address them.

In the next section, we transition to exploring language modelling for conversational agents. We review the main deep learning and neural networks approaches used throughout this work.

| Level | Description |
|---|---|
| 1 | Establishing contact |
| 2 | Speech recognition |
| 3a | Parsing: |
| 3aa | Recognising all words |
| 3ab | Determining syntactic structure |
| 3ac | Determining a *unique* syntactic structure |
| 3b | Resolving underspecification: |
| 3ba | Reference |
| 3bb | Tense, scope, presuppositions, lexical ambiguities, etc. |
| 3c | Contextual relevance, computing the rhetorical connection |
| 4 | Recognising speaker's intentions; evaluating resulting discourse structure |

Table 2.3: Fine-grained model proposed by Schlangen (2004) aligned with the communication levels from Clark's (1996) joint action ladder.

## 2.2 Conversational Agents

The aspiration to interact with computers through natural language has existed since the early days of Artificial Intelligence (AI) (Winograd, 1972). Mirroring human-like communication with machines would allow users to engage with a system intuitively, without extensive training. However, as previously discussed, human language is notoriously unspecified and context-dependent, keeping this aspiration out of reach to date.

In this thesis, we focus on interactions where natural language serves as the primary communication channel, while also examining human-agent interactions in a broader context that extends beyond language alone. We differentiate between *dialogue systems*, specifically designed to handle natural language interactions (e.g., chatbots), and *conversational agents*, which provide a more seamless interaction that may include non-verbal elements (e.g., gestures) and multi-modal capabilities (e.g., vision). Throughout, we generally use the term 'conversational agents' to encompass both, although at times we may concentrate on specific aspects (e.g., language modelling).

Chatbots are the prototypical early dialogue systems, driven by a large mix of rules and domain-specific handcrafted modules. For instance, if the user says *"Hello"* then the system would reply *"Hello, how are you?"*. These systems integrated multiple well-defined components with clear divisions, such as Natural Language Understand-

ing (NLU), Natural Language Generation (NLG) and dialogue management, to build an appropriate response.

However, due to the labour-intensive process of refining these rules and modules, these systems have mostly been replaced by data-driven methods trained on large-scale corpora. Technological advances in Machine Learning and subsequently, Deep Learning (Goodfellow et al., 2016), have made it feasible to train statistical-based systems, and these to integrate additional modalities (e.g., vision) to leverage context beyond language. As a results, individual components in these systems are no longer easily identifiable in a pipeline, leading us to refer to them as end-to-end systems.

Conversational agents communicate with users through natural language dialogues in a sequence of turns and fall into two overall categories (Jurafsky and Martin, 2000): goal-oriented, designed to aid in task-completion (e.g., setting a timer, asking for the weather); and chat-oriented, focused on free-form conversations for entertainment (e.g., chit-chat). Our work focuses on conversations with a particular goal (i.e., pick an item, arrange blocks), and hence we leave discussion of chat-oriented systems and their complexity as out of scope.

### *2.2.1 Deep Learning and Neural Networks*

Throughout this thesis, we explore conversational agents built on deep learning models trained on large-scale data. These models process inputs (e.g., raw data, sentences) through multiple layers of linear functions, iteratively generating higher-level representations that are distilled into a final output. This layered approach enables models to learn complex tasks (LeCun et al., 2015), such as object detection, that would be too challenging to design manually. Each layer contains a set of parameters that are adjusted and transformed during training to approximate an overall function $f_{layer}$ for a given input $x$:

$$\mathbf{f}(\mathbf{x}) = f_{\text{layer3}}(f_{\text{layer2}}(f_{\text{layer1}}(x))) \tag{2.1}$$

The model learns $f_{layer}(x)$ without requiring explicit specifications of what $f_{layer}(x)$ should model. This adaptability is a key advantage of deep learning over other approaches, as the layers, and thus the model itself, evolve to fit a complex function $f(x)$ during the learning process without traditional feature extraction. Another advantage is that thanks to these iterative layers, deep learning models have the potential to generalize

to previously unseen data (Bengio and LeCun, 2007). The internal architecture of these models can be represented as a graph of stacked layers, with multiple *hidden layers* between the input and output layers, by the interconnected networks in the human brain (see Figure 2.1). Thus, deep learning models are also referred to as neural networks. We will use the term *model* in this thesis.

### Feed-forward Networks

The simplest network architecture is a **Feed-Forward Network (FFN)**, which consists of a Perceptron (Rosenblatt, 1958) that takes an input $x$ and maps it to a linear function $f$ using weight $W$ and bias $b$: $f(x) = W\dot{x} + b$. Stacking layers with intermediate activation functions (also Multi-Layer Perceptron (Bengio et al., 2003)) allows the network to learn non-linear complex functions (Goodfellow et al., 2016), see Figure 2.1. In this thesis, we use several network architectures explained in detail below.

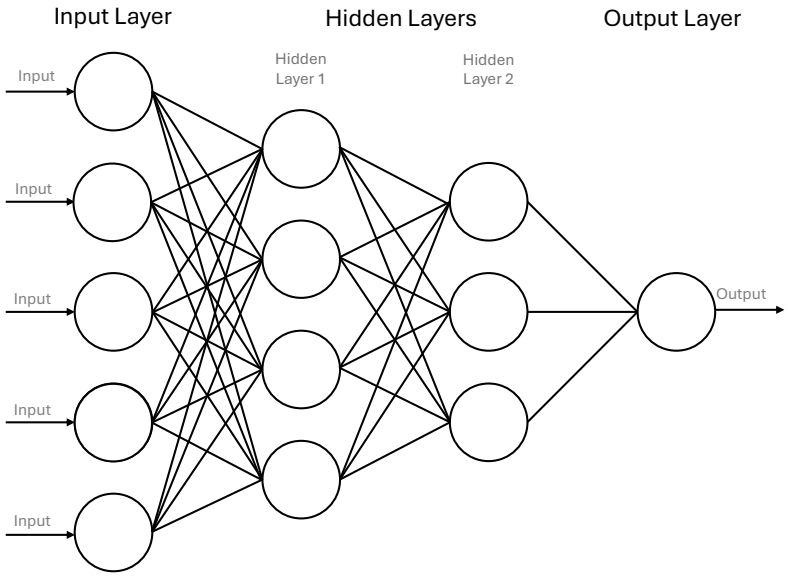



Figure 2.1: FFN architecture with two hidden layers.

### Recurrent Neural Networks

Early works on deep learning proposed RNNs (Rumelhart et al., 1988) to manage sequential data, such as text. Unlike simpler networks that process the input forward sequentially, this network transforms the input based on not only the learned parameters but also the previous output of the network itself. Thus, it processes the input sequentially as:

$$\mathbf{h_t} = f_h(h_{t-1}, x_t, \theta_h) \tag{2.2}$$

where $\mathbf{h}_t$ represents the hidden state of the network at time $t$, which is derived from applying $f$ to the previous hidden state, $\mathbf{h}_{t-1}$, the input $\mathbf{x}_t$ and the model parameters $\theta_h$ (weights $W$ and biases $b$ of each layer). RNNs compute back-propagation through time (Rumelhart et al., 1988) to estimate the parameters of $\theta_h$, learning to encode in the final hidden state $\mathbf{h_t}$ a summary of the input sequence. Once the sequence ends, we can use the final hidden state to obtain the output $f_o(h_t)$. See Figure 2.2 for an example.

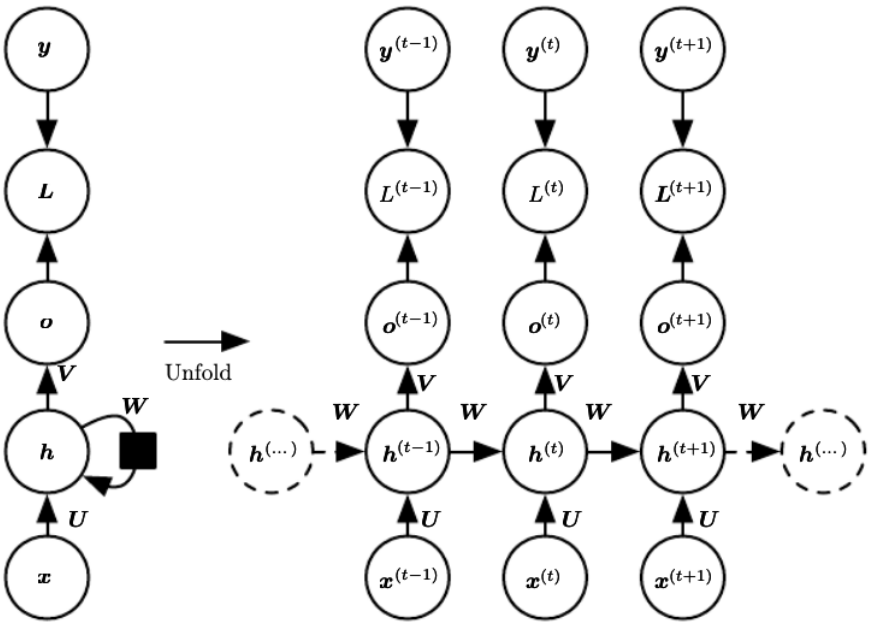



Figure 2.2: RNN architecture. The left side is repeated (unfolded) for each input token $x$ in the sequence to form the full network. Figure from Goodfellow et al. (2016)

RNNs are however prone to develop vanishing/exploding gradients (Bengio et al., 1994), where the back-propagation algorithm causes some layers to have excessively small or large parameters, causing negligible changes or instability across the network and thus preventing the network from learning. To avoid these issues, practical implementations of RNNs use intermediate units, such as Long Short-Term Memorys (LSTMs) (Hochreiter and Schmidhuber, 1997) that alter the flow of gradients and prevent this issue.

### Convolutional Neural Networks

Another type of network, CNNs (LeCun et al., 1989; Lecun and Bengio, 1995) rely on convolutions rather than matrix multiplication to extract high-level features from the

input. Each hidden layer applies a convolution over regions of the input, deriving relevant feature maps from local fields, which are then subsampled or pooled to summarize the information, see Figure 2.3. In essence, CNNs learn their own feature detector from the data (Lecun and Bengio, 1995), making them particularly effective for multi-dimensional data such as images.

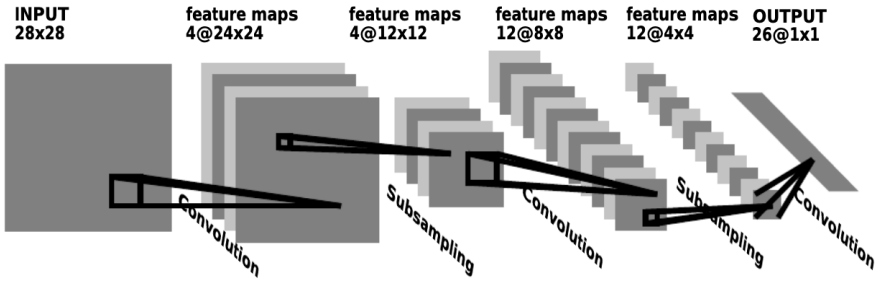



Figure 2.3: CNN architecture. Figure from Lecun and Bengio (1995).

### Transformers

The attention mechanism Bahdanau et al. (2015) enables networks to focus on relevant parts of the input when generating output sequences. Unlike RNNs, which condense the input into a summarised hidden state, attention aligns each word token to every other input token, thus partly mitigating issues like vanishing and exploding gradients.

Later, Vaswani et al. (2017) proposed the Transformer architecture, incorporating self-attention (Lin et al., 2017) to remove the network's dependency on recurrence, a computationally expensive process in sequential input processing. Instead of processing tokens one at a time as in RNNs, this architecture allows parallel processing of the input, capturing various relationships and dependencies across tokens.

The core component of the transformer layer is the Scaled Dot-Product Attention, which maps a set of queries $Q$ to key-value pair matrices $K$ and $V$. *Queries* represent the current token's search for relevant information, *keys* are 'labels' for all tokens that determine relevance and *values* contain the actual content to be aggregated. The output is the weighted sum of the values, with weights determined by the similarity between queries and keys:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

During training, the network learns to disregard irrelevant information, instead focusing on capturing complex dependencies across the input and further mitigating the vanishing gradient problem by scaling to the dimension of the keys, $\sqrt{d_k}$.

Each transformer layer combines the self-attention mechanism (multi-head attention) with a fully connected feed-forward network, residual connections to the input (He et al., 2016) and a layer normalisation (Ba et al., 2016), see Figure 2.4, which shows how the queries, keys and values interact within the attention mechanism. We distinguish between three transformer structures: encoder-decoder, encoder and decoder.

**Encoder-Decoder Transformer** This architecture is the default proposed by Vaswani et al. (2017), where the input is transformed into an output sequence through a set of encoder and decoder layers, similar to RNNs and LSTMs. It is highly suitable for a wide variety of tasks, such as machine translation or summarisation. Examples of models with this architecture include BART (Lewis et al., 2020) and T5 (Raffel et al., 2020).

**Encoder-Only Transformer** Unlike the encoder-decoder, this variant of the architecture only encodes an input into a representation, which can then be used for tasks like classification or extracting embeddings from input tokens (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)).

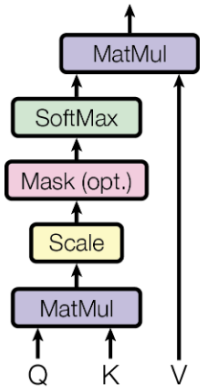
**Decoder-Only Transformer** This transformer generates output sequences by attending only to prior context (previous outputs), making it unidirectional. Decoder-only transformers are well-suited for text generation tasks (e.g., GPT-like models (Radford et al., 2018; Radford et al., 2019)) but cannot incorporate bidirectional context from the input.

### The Learning Loop

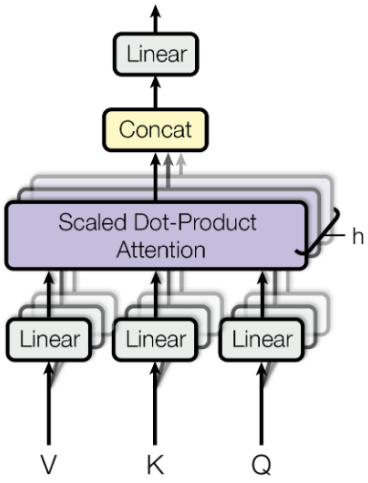
Deep learning's advantage over other methods relies on the efficacy of its training algorithms, allowing layers to learn patterns from large amounts of data. The core idea behind training deep neural networks is the following:

1. Pass an input sequentially through all the network layers to produce an output; this step is known as the *forward pass*.

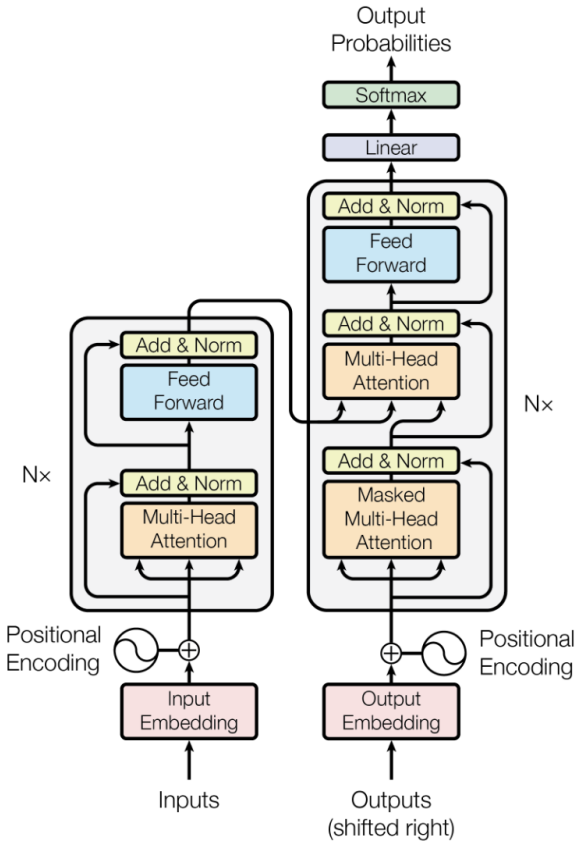
2. Measure the distance or difference between the predicted output and the expected value according to a predefined error or *loss function*.




(a) Scaled dot-product attention




(b) Multi-head attention




(c) Transformer encoder (right) and decoder (left).

Figure 2.4: Transformer architecture components. Figures from Vaswani et al. (2017). *Queries Q* seek relevant information, *Keys K* determine relevance and *Values V* provide the content to aggregate.

3. Starting from the output layer, compute each parameter's contribution to the final error and adjust the weights according to the gradient of the loss function. This process effectively propagates the errors backwards from the output layer through the hidden and then input layers, and is called *backpropagation* (Rumelhart et al., 1986).

To speed up computation during training, we repeat this process in batches of examples from the training data, averaging the losses over these examples. We describe these steps below but refer to Jurafsky and Martin (2024) for further information.

**Loss Function** To calculate how close the output of a network is to the expected or gold output, we use a loss function that maximises the log probability of the gold (correct) labels in the training data for a given input. This function ultimately depends on the task and network architecture, but the most commonly used function is the negative log-likelihood loss or Cross-Entropy (CE) loss:

$$\text{Loss}_{CE}(\hat{y}, y) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{2.4}$$

where $C$ is the number of classes, $\hat{y}$ is the predicted output probability for class $i$ and $y$ is the gold or expected output.

**Backpropagation** Backpropagation (Rumelhart et al., 1986) optimises the loss function by computing gradients after the forward pass. The gradient descent algorithm adjusts the layer parameters so that future forward passes predict outputs closer to the gold output. The most commonly used function is Stochastic Gradient Descent (SGD), which randomly samples $m$ examples from the training data in batches (e.g., input $X = \{x_1, x_2, ..., x_m\}$), averaging losses across these examples:

$$\theta := \theta - \eta \cdot \frac{1}{m} \sum_{i=1}^{m} \text{Loss}_{CE}(\hat{y}_i, y_i) \tag{2.5}$$

where $\theta$ represents the network layer parameters, and $\eta$ is the learning rate to control the magnitude of each update. We usually choose a learning rate value that avoids small updates, which would slow down training, or too large, leading to uncontrolled updates. We apply SGD to each layer using backpropagation, in which the computation is broken down into separate operations over a graph. This passes the gradients from the final nodes (output) to all the nodes in the graph. Refer to Jurafsky and Martin (2024, Section 7.5) for additional details. There are many SGD variants, with Adam (Kingma and Ba, 2015) being the most common optimiser for language modelling.

### *2.2.2 Training Approaches*

This section provides information on common training approaches for deep neural networks. In this thesis, we use models trained with self-supervised learning, which we further train with supervised or reinforcement learning.

### Supervised Learning

When we have a correct output $y$ for a given input $x$, we can use the previously discussed algorithms to train a neural network for that task. This approach, called supervised training, requires labelled data or the gold outputs to learn the network parameters that best approximate $y$. For instance, to learn to detect dogs in images, you need a dataset of labelled images with 'dog' and 'no dog'.

### Self-Supervised Learning

Deep learning models, particularly nowadays, are trained on the premise that a particular word in a sentence can serve as the gold output or learning signal (Bengio et al., 2003). This method is called self-supervision and enables models to learn to predict a word within a word sequence without manually labelled data, in contrast to supervised approaches.

Transformers in particular exploit this self-supervision to train on massive datasets, developing strong general language modelling capabilities that can be transferred to other tasks. We will discuss pre-trained language models in Section 2.2.3.

### Reinforcement Learning

A different approach is to learn what to do based on observations of the world and a reward, known as RL (Sutton and Barto, 2018). In this case, the training does not need labels (i.e., supervised learning) and maximises rewards during training episodes. RL is well-suited for virtual agents in simulated environments, where they can learn behaviours over long simulations, such as navigation or object manipulation.

Initially from general AI such as robotics, it has recently gained popularity with the introduction of more generalisable algorithms and RLHF for conversational agents. We will explore RL in Chapter 3 and Section 6.6 to align models towards particular behaviours.

In RL, the agent or model does not have prior knowledge of the reward associated with each action or the consequences on the environment, encouraging it to make mistakes and learn from them. Given that the action and search space may be infinite, the agent must develop an optimal policy that combines 1) *exploitation*, to select the action that maximises rewards; and 2) *exploration*, to perform actions with unknown rewards or

outcomes (i.e., learn). Thus, there is a dynamic relationship between the model and the world, as seen in Figure 2.5.

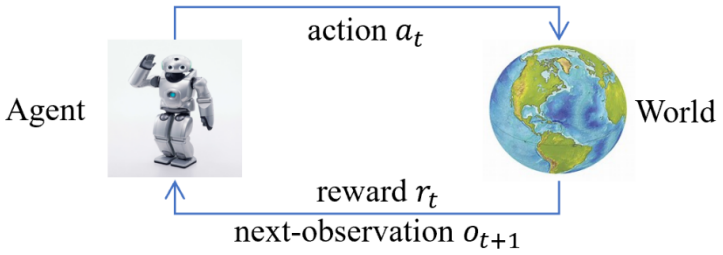



Figure 2.5: Interaction of an RL agent with the environment from Gao et al. (2019). The agent explores the environment, executing actions and receiving rewards and observations about the world, which are then used to update the its policy and improve the agent's performance. This cycle repeats until the agent learns an optimal policy (optimal action-selection strategy).

RL has the advantage of requiring no initial training data, allowing the agent to explore on its own and potentially achieve better outcomes than with supervised or unsupervised learning in a process similar to human learning itself. However, RL requires a large number of trial and error, which may not always be feasible. For instance, a robot could risk damaging expensive hardware, or a conversational agent may explore tactics that lead to poor user experiences. While simulated environments can be used for initial training, they often lack the complexity needed for effective real-world application.

The model learning process (weights and parameters) is highly dependent on the final implementation details, such as the RL algorithm used (e.g., Q-Learning, PPO). We provide some details below of the ones used or relevant to this work:

- **Q-Learning** (Watkins and Dayan, 1992): a value-based algorithm that learns a function for each action-state pair, estimating how rewarding a state is based on expected cumulative rewards. For instance, Mnih et al. (2013) trained a convolutional network to play Atari games with Q-learning.

- **Proximal Policy Optimisation (PPO)** (Schulman et al., 2017): is a policy-based algorithm that uses two networks working together, actor-critic, to predict actions and estimate values respectively to learn an optimal policy. We will use Proximal Policy Optimisation (PPO) in Chapter 3.

RL struggles in general with language tasks due to its sparse rewards (language has no reward signals, unlike doing the correct action in a simulated environment) and the lack of clear objectives or goals (i.e., what makes a good response in a dialogue). In a simple virtual environment, this may be easy to define (e.g., distance to a goal), but it is harder to quantity or design for complex tasks, such as conversational agents.

Despite these challenges, recent works have developed new approaches to harness RL's capabilities for language tasks. RLHF addresses the limitations of traditional RL by transforming abstract human preferences into concrete reward signals that language models can optimise against.

### Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019) aligns models more closely with human preferences (Stiennon et al., 2020; Bai et al., 2022a) by overcoming the sparse reward problem. Instead of relying on arbitrary reward functions, RLHF leverages human evaluations to create feedback signals, bridging the gap between technical capability and human judgment. The feedback from human annotators acts as a proxy for alignment, steering models towards generating more 'human-like' output. This approach typically requires models with strong language modelling capabilities, but RLHF has been remarkably effective for training the most capable models available, including GPT-4 (OpenAI, 2024).

The core idea behind RLHF is to maximise cumulative rewards whilst using a KL divergence to constraint updates. KL divergence measures the relative entropy between two distributions and, in this case, prevents models from hacking rewards in RL, where reward scores would be high but human evaluations would be low (Stiennon et al., 2020; Touvron et al., 2023b).

As shown in Figure 2.6, RLHF requires multiple models working together, including a reward model (usually previously trained with supervised learning) and two language models: one that is fine-tuned and an identical, frozen Language Model (LM) for calculating KL divergence during training. In the standard RLHF training loop, text inputs pass through both the initial Language Model (LM) and the LM being tuned (initially identical). The outputs are used to calculate KL divergence between them, creating a penalty that prevents the tuned model from diverging too far from the initial model. A separate

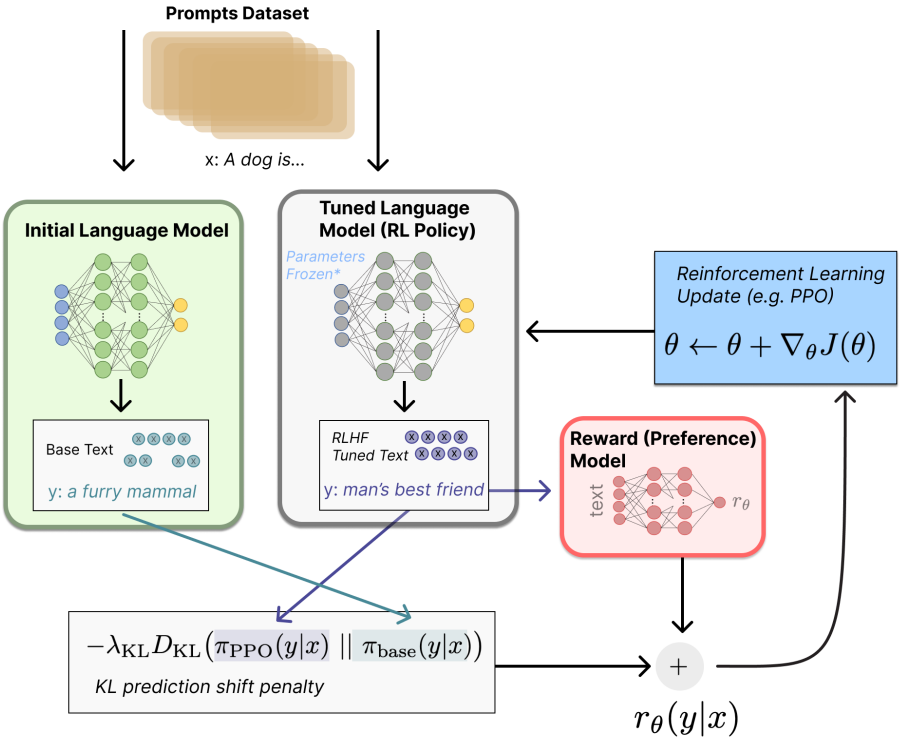



Figure 2.6: RLHF diagram from (Lambert et al., 2022) showing the training process. In the training loop, inputs pass through both models to calculate the KL divergence penalty, while the reward model evaluates the tuned model's outputs. These combined signals guide the optimisation process, which only updates the tuned model while keeping the rest frozen.

reward model evaluates the tuned LM's outputs, and these scores (combined with the KL penalty) drive the RL optimisation, which updates only the tuned LM while keeping all other components frozen.

Depending on the implementation, all models but one may be frozen, or several models are trained jointly (e.g., Bai et al., 2022a, updates reward and policy models simultaneously). In some cases, the feedback may come from another language model instead of human annotators. Anthropic's Claude[5] for instance, has been successfully trained on automatically-generated AI rather than human feedback in an approach known as Reinforcement Learning from AI Feedback (RLAIF) (Bai et al., 2022b).

While standard RLHF effectively uses preference data, its reliance on training a separate reward model can introduce complexity and instability. We discuss a variant of RLHF used in Section 6.6 below, DPO, which addresses this by bypassing the explicit reward modelling step.

[5]See https://www.anthropic.com/claude/sonnet [Last accessed: 7 Nov 2024]

**Direct Preference Optimization (DPO)** (Rafailov et al., 2024) It offers a more direct method for aligning language models with human preferences compared to RLHF, as it bypasses the explicit training of a separate reward model. Instead, it leverages the relationship between reward functions and optimal policies, enabling direct optimization of the model policy using preference data. This data typically consists of pairs of responses to a given prompt, where one response is preferred over the other (e.g., chosen by a human annotator). DPO uses a specific loss function encouraging the model to increase the likelihood of generating the preferred responses and to decrease the likelihood of the less desirable ones.

This approach simplifies the RLHF pipeline, reducing computational costs and potentially improving stability compared to the multi-stage complete RLHF process, which involves training a reward model and then fine-tuning the language model with RL. We provide a visual comparison of the streamlined nature of DPO relative to RLHF in Figure 2.7. We will explore both RLHF and DPO in Section 6.6.

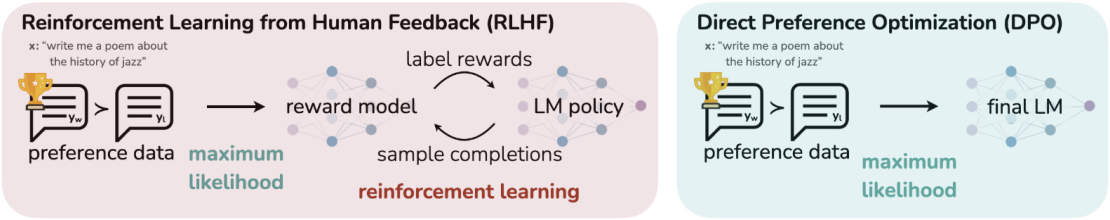



Figure 2.7: Comparison between the RLHF and DPO pipelines, adapted from (Rafailov et al., 2024). The diagram highlights the key difference between the two: RLHF requires a separate reward model that scores model outputs, while DPO directly optimises the model policy using preference data. In some cases, RLHF also requires re-training the reward model, converting it into a more complex multi-stage process that is more difficult to optimise. DPO simplifies this process in the cases where preference data is available.

### *2.2.3 Pre-Trained Large Language Models*

The introduction of the transformer architecture (Vaswani et al., 2017) and self-supervised learning significantly improved language models' capabilities. Since self-supervised learning does not require labelled data, models can easily be trained with extensive text data readily available on the Internet. This enabled three primary methods to train models:

- **Causal Language Modelling:** models train to predict the next token from an input

sequence in an autoregressive manner. These models typically follow a decoder-only (also *autoregressive decoder*) architecture and are well-suited for text generation, e.g., GPT models (Radford et al., 2018; Radford et al., 2019).

- **Masked Language Modelling:** models train to predict tokens that have been masked (i.e. by a special [MASK] token) in the input, learning to attend to the entire sequence. These models usually have an encoder-only architecture, e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019).

- **Sequence-to-Sequence:** models train to first encode an input sequence, then predict the output as a reconstruction of the input. This method is closer to the LSTMs architecture and is implemented with an encoder-decoder transformer setup, e.g., BART (Lewis et al., 2020) and T5 (Raffel et al., 2020).

Training models with these approaches on billions of words and input sequences (e.g., Wikipedia, Reddit, social media) enables them to develop strong language representations that are universal across many tasks. Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) are two early examples of learning word embeddings to transfer to other tasks. More recently, instead of training models from scratch (i.e., freshly initialised parameters), we reuse previously trained models that have developed strong base language modelling capabilities, *pre-trained language models*, and can later be further fine-tuned for specific downstream tasks. These pre-trained models require much less data to be trained in new tasks, being more data and computationally efficient.

The size of these pre-trained models has also increased exponentially, for instance, BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), early pre-trained models, have 340M and 1.5B respectively; whereas the latest open models like Llama 3 (Dubey et al., 2024) have 90B. Notably, the larger models, referred to as *LLMs* henceforth, have started showing considerable generalisation capabilities, enabling the use of zero-shot *prompting*, where some task instructions are sufficient for models to adapt and perform well on the task (Brown et al., 2020).

Other capabilities have emerged, such as In-Context Learning (ICL) (Brown et al., 2020; Liu et al., 2023), where models are conditioned with task instructions and a few examples to generate text predictions, without task-specific training in zero and few-shot settings (Wei et al., 2022; Chowdhery et al., 2022). Thus, LLMs are increasingly used as

all-purpose models for tasks beyond text generation, such as classification and regression (Zhu and Zamani, 2024; Salemi et al., 2024) by simply parsing the output.

### Efficient Training Methods

Due to the large size of the latest models, it is often infeasible to train them from scratch and then fine-tune them for a specific task. It is more efficient to reuse the already developed embeddings and tweak the parameters in the last few layers to obtain what we want. Two methods in particular are relevant to this work, which we discuss below.

**Low-Rank Adaptation (LoRA)** Instead of training all model parameters during fine-tuning, a small layer on top of the last model layer is added, called an adapter layer, and is the only one that updates its parameters. This technique relies on strong pre-trained model language capabilities, adapting models to novel downstream tasks without computationally-demanding fine-tuning. This process is Low-Rank Adaptation (LoRA) (Hu et al., 2021) and the original model layers are *frozen*, so no parameters or weights are trained.

**Quantisation Low-Rank Adaptation (QLoRA)** Another relevant training method is Quantisation Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023), which combines LoRA with *quantisation* (Dettmers et al., 2022a) to reduce the memory requirements needed to load and train LLMs. **Quantisation** reduces the precision of a model's weights and activations from floating-point numbers to smaller integer values (e.g., from 32-bit floats to 16-bit integers). It considerably reduces the memory and computation requirements to train large models, at the cost of losing accuracy in the model's predictions and a more challenging training, as optimisers like Adam (Kingma and Ba, 2015) expect floating-point numbers. QLoRA in particular uses 4-bit quantisation, reducing the precision of the model parameters and thus significantly decreasing memory usage. By lowering the precision, we are able to both load LLMs in less powerful hardware and fine-tune more efficiently without significantly impacting model capabilities (Dettmers et al., 2022b).

### 2.2.4 Multi-Modal Models

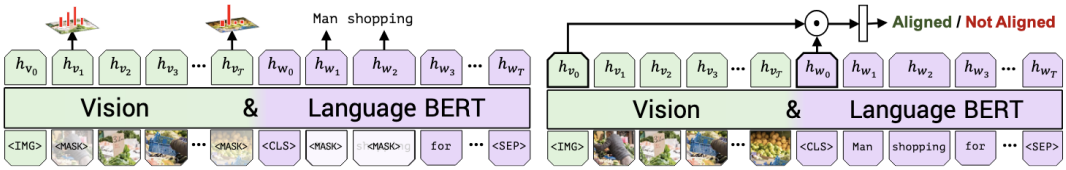
Conversational agents have also expanded beyond language in recent years, incorporating additional modalities. Early works on multi-modal, end-to-end neural networks introduced architectures that combined CNNs for processing visual or spatial inputs with LSTMs for encoding language (e.g., Bisk et al., 2016; Mei et al., 2016; Misra et al., 2017; Misra et al., 2018; Suhr and Artzi, 2018). Similar to LLMs, the introduction of the transformer architecture (Vaswani et al., 2017) and self-supervised learning have significantly improved multi-modal model's abilities across a variety of tasks.

To develop robust representations, models are pre-trained on tasks that combine multiple modalities. This work uses extensively vision and language models, while occasionally experimenting with additional information from 'adjacent' modalities to vision (e.g., spatial or object relationships), so we will only cover works that integrate these modalities.

Vision-and-Language Models VLMs process inputs from both visual and language modalities, aligning them into a single unified representation. Initial VLMs relied on attention across modalities (e.g., LXMERT (Tan and Bansal, 2019) or VilBERT (Lu et al., 2019)) and pre-trained on a range of visual-language alignment tasks. These tasks aim to mirror the masked language modelling approach used in BERT (Devlin et al., 2019), and include image-caption matching, Visual Question Answering (VQA) or REC (see Figure 2.8 for two commonly-used pre-training tasks). To encode visual inputs, these models would first extract RoI features from image regions using object detection models, for example ResNets (He et al., 2016), followed by a linear visual encoder. The pre-trained models transfer their learned capabilities to downstream tasks by fine-tuning them on task-specific datasets in an end-to-end manner, adapting to specific tasks and improving their performance.

Nowadays, VLMs have simpler architectures, typically using vision transformers (Dosovitskiy et al., 2021) to encode visual inputs. Both vision and language encodings are projected into a shared space using a Multi-Layer Perceptron (MLP) (Bengio et al., 2003), enabling stronger integration between modalities. Figure 2.9 shows an example of such architecture. Idefics2 (Laurençon et al., 2024) and LLaVA (Liu et al., 2024b) are notable examples of these VLMs which we use in Chapter 6.

The following sections introduce the collaborative tasks that have been previously




(a) Masked multi-modal learning (b) Multi-modal alignment prediction

Figure 2.8: Pre-training tasks for vision-language alignment from Lu et al. (2019). (a) Masked multi-modal learning masks 15% of words and image regions, which the model has to reconstruct using the outputs of either the detection model (e.g., ResNet) or text (same as in BERT) as the supervision signals for each modality. During training, it minimises the KL divergence between the object detection model's outputs and the reconstructed regions. (b) Multi-modal alignment prediction consists of a binary classification of whether an image-text pair are aligned with each other or not (whether the text describes the image).

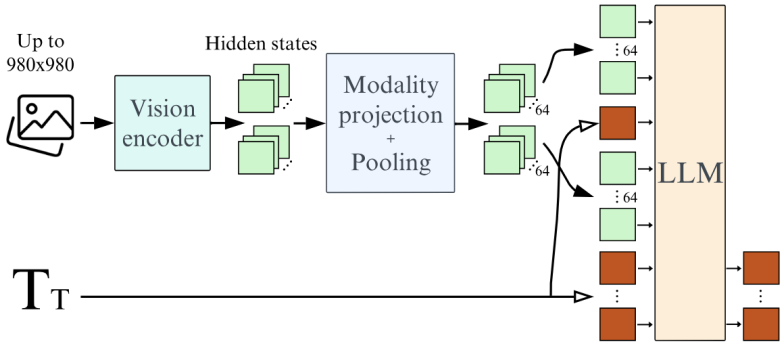



Figure 2.9: Idefics2 architecture from Laurençon et al. (2024). Instead of relying on separate object detection models (e.g., ResNet), it encodes the visual inputs and projects to the same embedding space as the language, concatenating all together before the final LLM backbone.

explored with vision and language models, and discuss how these inform our approach to addressing the research questions.

## 2.3 Situated Dialogue

Throughout this work, we particularly focus on *situated* environments, where the human and agent are both present in a shared environment and are actively engaged. This may be a physical space (i.e., co-located) or a virtual one, such as most of the work here, and usually involves the notion of close proximity between human and agent with shared visual perspectives. This contrasts with remote environments (e.g., remotely operated offshore robots (Hastie et al., 2018)), where the requirements for collaboration are different (see discussion in Chiyah Garcia et al., 2020a) and an overseer dialogue system may be more

suitable (Chiyah Garcia et al., 2018a; Hastie et al., 2017; Robb et al., 2018; Lopes et al., 2019). Situated robots are inherently embodied in a shared environment, thus they need to adapt to evolving physical and social context, such as surrounding objects or agent/human actions, and conform to social or safety norms.

Situated dialogue inherently intertwines the communicative and symbol grounding processes discussed earlier. The shared environment provides rich contextual resources that participants leverage to achieve mutual understanding (Clark, 1996). A key mechanism enabling this is *joint attention*, the ability of participants to coordinate their focus on the same objects or events in the environment (Tomasello, 1995). Thus, establishing and maintaining joint attention, often through cues like gaze or pointing, is crucial for successful reference and grounding (Clark and Brennan, 1991). For instance, pointing at an object while referring to it uses communicative grounding to direct joint attention, facilitating the symbol grounding process by clearly linking the linguistic expression to its object in the shared environment.

An agent following natural language instructions in such settings needs to fully comprehend both their meaning and its surrounding environment (Misra et al., 2017). This requires grounding language not just abstractly, but specifically within the current context, directly addressing the symbol grounding problem (Harnad, 1990) by linking words to perceived entities and actions. Since grounding involves perception and action (Barsalou, 1999) and cannot rely solely on text (Bender and Koller, 2020), many multi-modal models incorporate visual or world-state information. This integration aims to bridge language, perception and action, spurring grounding and connecting it with embodiment and cognition (Barsalou, 2008), as explored in works like Mei et al. (2016).

While agents integrating text and vision demonstrate improved generalisation to novel tasks, their capabilities remain limited without the ability to actively experience and modify their surrounding environment (Bisk et al., 2020). Achieving true human-agent collaboration requires agents that not only understand their surroundings but can also reason and cooperate via language grounded in shared experience. We now discuss several tasks involved in situated dialogue to explore these capabilities below.

### 2.3.1 Natural Language Grounding

This thesis explores miscommunications in multi-modal interactions, specifically within situated environments. In this section, we explore *grounding* as the connection of a particular natural language sentence to an action or object, which is a crucial capability for robust conversational agents operating across diverse scenarios. Situated dialogue requires agents to integrate several core capabilities.

Firstly, an agent needs to perceive the shared environment using sensors (e.g., camera), identifying relevant objects with their properties and understanding the relationships between them. For example, an agent in ALFRED (Shridhar et al., 2020) needs to perceive the environment using a camera, identify the location of objects like an apple and a knife, and the layout of the scene, eventually understanding that the apple is on the table and the knife to cut it is on the kitchen counter.

An agent must also interpret the user's utterances, linking it to the perceived world and determining which specific object or location is being referenced (e.g., understanding which object the user refers to when they say 'the red sphere' as in CLEVR (Johnson et al., 2017)). Furthermore, an agent may need to maintain context and manage the conversational flow, producing appropriate responses or planning and executing actions when required. For example, planning and following a set of directions in Visual Language-Navigation (VLN) tasks such as Room-to-Room (Anderson et al., 2018b)[6] or determining when to act or ask a clarification question (see Thomason et al., 2020; Roman Roman et al., 2020; Chi et al., 2020; Shi et al., 2022).

These individual steps or tasks are often addressed simultaneously in embodied AI research, as they are integral aspects of human-agent collaboration in situated dialogues. Within this context of interconnected tasks, we identify two key research areas for our work: resolving referring expressions (a core component of both communicative and symbol grounding) and following instructions (which integrates understanding and grounding in general). While other tasks like scene understanding, dialogue management and response generation are essential components of a complete situated dialogue agent, our primary focus lies on these two areas.

[6] Also see TOUCHDOWN example in Figure 2.12.

### 2.3.2 Referring Expression Comprehension

Referring expressions represent a fundamental challenge in bridging language and perception, requiring an agent to associate a phrase in natural language with its corresponding physical object or entity in the environment. This task is closely tied to solving the symbol grounding problem (Harnad, 1990), as it requires mapping abstract symbols (words and phrases) to concrete perceptual data. For instance, interpreting the utterance *"What is the price of that blue jacket?"* requires the agent to understand the lexical meaning and to visually identify (and ground) the specific referent denoted by the expression 'that blue jacket' within the current context.

This process can become significantly complex, as referring expressions might target single objects (e.g., 'the red t-shirt on the table'), multiple objects ('those shirts') or even objects defined by their relative relationships to others ('the t-shirt to the left of the wardrobe'). Furthermore, and as previously discussed, descriptions are typically underspecified and may use pronominal references (e.g., 'the red one') or require contextual inference ('the one I mentioned earlier'). These factors may lead to ambiguity, as the same expression could match multiple candidate objects which agents have to resolve.

A notable set of datasets to learn Referring Expression Comprehension (REC) are RefCOCO (Yu et al., 2016a) and its variants (RefCOCO+ (Yu et al., 2016a) and RefCOCOg (Mao et al., 2016)), which are typically first used to pre-train visual-and-language models prior to more complex tasks. Figure 2.10 shows an example of a REC task, where a model has to find the bounding box of the object referred by the expression. VQA and visual dialogue are notable downstream tasks of REC in situated dialogues, as they involve questions and statements about objects depicted in images, which require robust grounding capabilities to resolve (e.g., see works by Antol et al., 2015; Goyal et al., 2017; Das et al., 2017; de Vries et al., 2017; Kottur et al., 2019).

We review relevant works in REC in Chapter 4, where we introduce a model designed to learn and resolve referring expressions.

**Generating Referring Expressions** Closely related to REC is the task of Referring Expression Generation (REG), also important in situated dialogues. REG involves producing an unambiguous natural language description that allows a listener (human or another agent) to identify a specific target object or location within a given context.

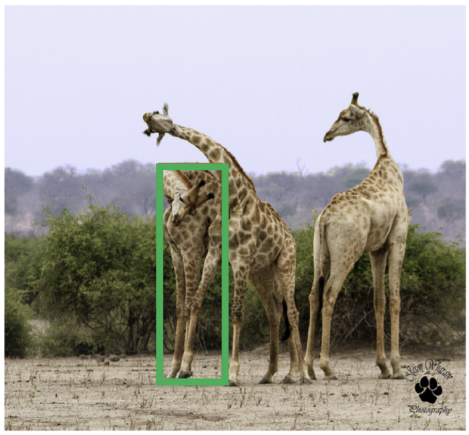



Figure 2.10: Example referring expressions for the giraffe outlined in green for all the RefCOCO datasets. Figure from Yu et al. (2016a).

This task presents several challenges, as the generated expression must be discriminative enough to single out the target from other potential distractors in the scene, yet ideally remain concise[7]. This requires careful attribute selection, deciding which properties (e.g., colour, size, spatial relation) are the most salient and informative to identify the target object. Furthermore, the ordering of these attributes can impact clarity and naturalness, a problem extensively studied by works like (Dale and Reiter, 1995). Generating effective referring expressions often requires considering both the visual scene and potentially the listener's assumed knowledge.

Whereas REC focuses on interpreting a given expression to identify its referent, REG focuses on producing an appropriate expression given a target referent and its context. We do not address REG in this thesis and instead solely focus on interpreting referring expressions. Refer to Krahmer and Deemter (2019) for a comprehensive survey or other works (e.g., Kelleher and Kruijff, 2006; Gervits et al., 2021; Willemsen and Skantze, 2024; Willemsen et al., 2023).

### *2.3.3 Instruction Understanding*

Situated dialogue involves grounding language in the physical environment and the ongoing interaction. A core task is instruction understanding, which involves interpreting natural language instructions and mapping them to potential actions the agent can perform, while reasoning about its affordances within the environment. For example, the command "move the box" requires identifying the specific box, planning a navigation

[7]Underspecified utterances are cognitively less demanding than fully specified ones (Levinson, 2000).

path and executing the ‘move‘ action. This process often involves resolving ambiguities (e.g., “Which box?”) and grounding referring expressions (e.g., connecting “the box” to a specific visual object). This is closely related with instruction following, which encompasses both understanding and executing said instruction.

Situated dialogue agents must thus handle challenges such as managing dialogue context across turns, perceiving the environment, planning and executing actions and adapting to dynamic changes. Many instruction-following tasks combine visual perception with language instructions, spurring the development of multi-modal models capable of processing and integrating information from different sources. For example, works like Chen and Mooney (2011) used a corpus of navigation instructions to teach a model to interpret and follow them in a simple virtual world (see Figure 2.11); or Tellex et al. (2011) who learn to interpret natural language commands for robotic navigation and manipulation.

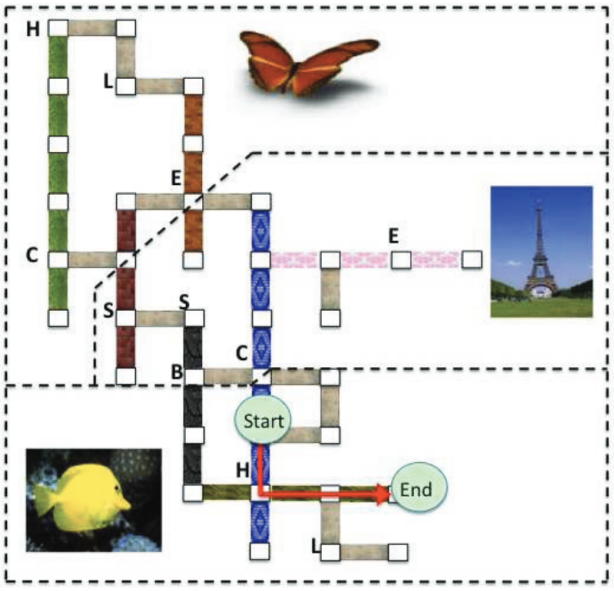



Figure 2.11: Route from a virtual world corresponding to the instruction “turn so that the wall is on your right side. walk forward once. turn left. walk forward twice.”. An agent has to interpret the instruction and navigate to the goal (End). Figure from Chen and Mooney (2011).

Often, the distinction between instruction understanding and REC (identifying the object or location mentioned) is not clear, and many tasks require handling both simultaneously. For instance, the instruction “pick up the red block” contains both an action (‘pick up‘) and a referring expression (‘the red block‘) that must be grounded visually.

Examples where these are intertwined include VLN tasks[8], such as Room-to-Room (Anderson et al., 2018b), which challenges agents to navigate realistic indoor environments following descriptions that reference specific landmarks and spatial relationships, and TOUCHDOWN (Chen et al., 2019), which involves navigating street view scenes to find a specific object based on detailed instructions, testing fine-grained grounding. Robotic manipulation benchmarks also highlight this interplay, like VIMA-Bench (Jiang et al., 2023b), featuring diverse tabletop scenarios that demand compositional understanding of language commands (e.g., rearranging objects based on properties), and CLIPort (Shridhar et al., 2022), where agents learn to perform various simulated manipulation tasks (e.g., packing, stacking) by grounding language instructions with visual inputs. Figure 2.12 shows examples from the TOUCHDOWN and VIMA-Bench datasets, where REC and instruction understanding are intertwined.

Beyond interpreting single commands, situated dialogue systems must also manage dialogue context across turns, ask clarification questions for ambiguous or incomplete instructions (e.g., "Which box do you mean?"), and adapt to dynamic environmental changes. These capabilities are crucial for effective human-robot collaboration. Chapter 3 provides a more extensive review of prior works and datasets in these related areas.

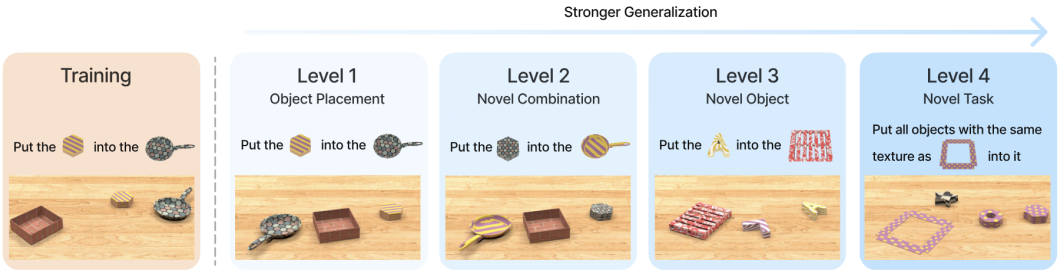
### *2.3.4 Collaborative Grounding*

Real-world situated dialogues often requires multi-turn interaction, similar to how humans engage in dialogue to clarify, coordinate and complete tasks collaboratively. Traditional instruction following or referring expression resolution tasks focus on single, complete sentences that are clear and interpretable, as most of the works discussed so far.

More recent works have explored grounding in dialogues (instruction-following or REC) in a variety of scenarios. For example, visual dialogue tasks (Das et al., 2017) involve sequences of questions and answers about an image (e.g., *"How many tigers are in the image?"* $\rightarrow$ *"3"* $\rightarrow$ "What are the tigers doing?" ...). Datasets like Talk The Walk (de Vries et al., 2018) feature navigation dialogues where participants can freely ask questions and coordinate (e.g., *"Is that a shop or restaurant?"* $\rightarrow$ *"A clothing shop"* $\rightarrow$ ... *"Should I go to the corner?"*). Similarly, TEACh (Padmakumar et al., 2022) and DialFRED (Gao et al., 2022) extend ALFRED (Shridhar et al., 2020) to allow

[8]Refer to Gu et al. (2022) for a survey on VLN.


(a) TOUCHDOWN example from Chen et al. (2019). The agent has to navigate a photo-realistic environment until reaching the goal (Touchdown the bear).




(b) VIMA-Bench example from Jiang et al. (2023b). The agent has to interpret and execute instructions about novel objects in a pick&place manipulation task.

Figure 2.12: Examples from datasets that combine instruction-following and REC.

communication between a commander and a follower agent to ask clarifying questions, see Figure 2.13 (e.g., Follower: *"I don't see the bowl"* → Commander: *"It's on the bottom shelf"*).

However, the dialogues in many of these benchmarks are collected under constrained or simulated conditions. For example, TEACh and DialFRED were collected asynchronously, potentially affecting the spontaneity of clarification requests or real-time negotiation common in human interaction. In fact, in some datasets (e.g., Visual Dialog (Das et al., 2017)), only a small fraction of responses rely on the dialogue context (Agarwal et al., 2020), limiting any meaningful analysis of communicative grounding.

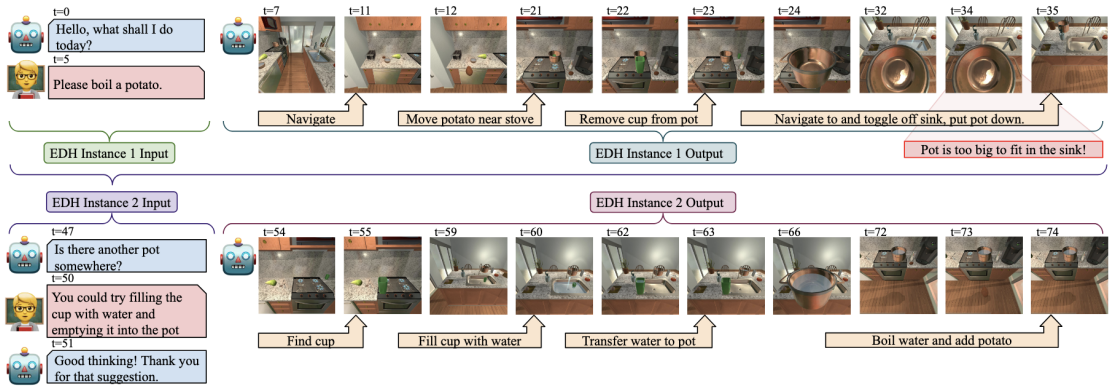



Figure 2.13: Example dialogue from TEACh. The follower and commander engage in a short collaborative dialogue to reach the goal. The dialogue was collected asynchronously, and most of the follower's questions are due to a knowledge gap (e.g., not finding a pot), rather than a miscommunication or unclear instructions. Figure from Padmakumar et al. (2022).

Furthermore, setups like GuessWhat?! (de Vries et al., 2017) often implicitly filter out complex coordination phenomena like miscommunications, interruptions and repairs, focusing instead on simpler question-answering patterns. Even embodied AI benchmarks (e.g., Shridhar et al., 2020; Gao et al., 2023b), aiming to emulate real-world noise, often significantly simplify ambiguity (Suglia et al., 2024), allowing models to resolve choices (like distinguishing a red vs. green spoon) as a sequence of actions, instead of requiring more elaborate clarification as in human dialogues.

This focus on structured Q&A or knowledge gap resolution means that the challenge of handling inherently incomplete, underspecified (Pezzelle, 2023) and ambiguous language, common in human collaboration, often remains insufficiently addressed. For instance, interpreting the instruction "Turn and go with the flow of traffic..." from the top of Figure 2.12 requires contextual reasoning beyond simple object identification. An agent might make the wrong turn, leading to task failure, a common issue in complex tasks like ALFRED (Shridhar et al., 2020) in unseen scenarios. Few works even consider the collaborative nature of language and the inherent underspecification within human dialogues.

Early work like the HCRC Map Task Corpus (Anderson et al., 1991) addresses instruction understanding in an abstract navigation task, involving dialogues that refer to landmarks and commonly employ coordination phenomena (e.g., acknowledgements). CEREALBAR (Suhr et al., 2019) uses a simulated world with different leader-follower perspectives to investigate referring expressions, proposing an approach for models to

reason and recover from errors between instructions (similar to Misra et al. (2018)); however, the follower cannot reply through language and non-linguistic actions are the only form of feedback. Yet, overall, research addressing rich dialogue phenomena, including miscommunications and repairs, remains limited, overlooking fundamental aspects of dialogue cooperation.

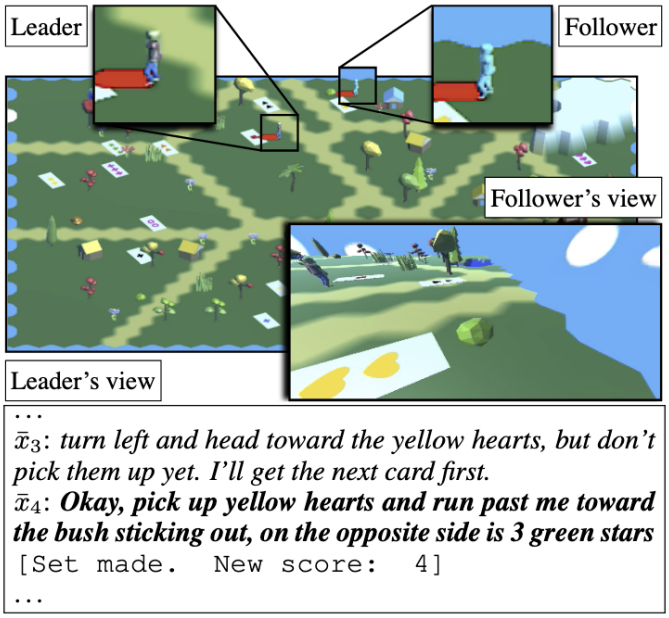



Figure 2.14: Example from CEREALBAR. There is no dialogue, just a sequence of instructions that the follower has to execute in the environment. Figure from Suhr et al. (2019).

### 2.3.5 Miscommunications in Situated Dialogues

As we have seen, most of the works discussed so far have collected data or built models in ways that simplify or prevent miscommunications, sometimes forcing human clarifications instead. To model truly situated dialogues, however, we need to embrace the full range of human coordination phenomena, including miscommunications and repairs, enabling agents to learn robust handling strategies to recover from them in situated dialogues (Marge and Rudnicky, 2015).

The HCRC Map Task Corpus (Anderson et al., 1991) serves as an example rich in such coordination phenomena, although its size (128 dialogues) limits its utility for training modern language models. Other works have also attempted to capture the collaborative nature of communication in situated dialogues. For instance, Tellex et al. (2014) investigated an agent that asks for help when it cannot recover autonomously, generating

targetted requests for clarification that would increase the effectiveness of the intervention. Chai (2018) explored learning a joint representation for symbol and communicative groundings based on dialogues where a human teaches a robot agent to perform a task.

Narayan-Chen et al. (2019) released a dataset of dialogues for an instruction-following task, where the human and the agent have to collaborate to build structures in the popular game Minecraft. These dialogues naturally contain miscommunications and repairs (see Figure 2.15), yet the authors did not explore them in detail and instead focused on the task of instruction generation. Notably, and closely related to this thesis, Thomason et al. (2015) proposed a dialogue agent that learns to interpret instructions and actively resolve ambiguities in human-robot dialogues. Their approach uses a semantic parsing model that learns incrementally from interactions, addressing the inherent uncertainties of situated communication. Finally, Madureira and Schlangen (2023) extended the CoDraw dataset (Kim et al., 2019) with annotations of clarification requests and repairs. They explored the tasks of determining when to make a clarification request and how to detect them, but did not focus on interpreting or repairing these requests as we do here.

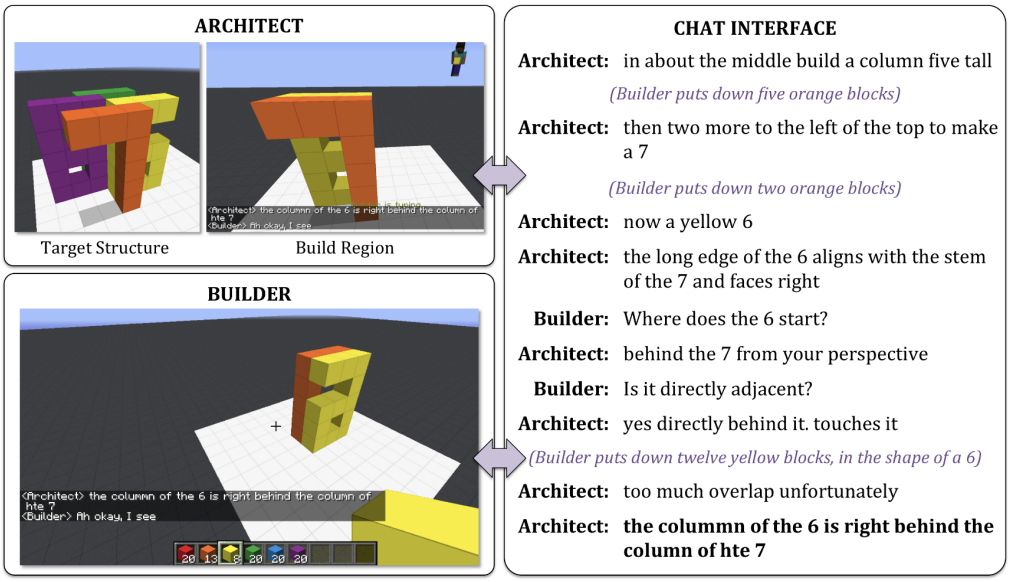



Figure 2.15: Figure from Narayan-Chen et al. (2019) with an example dialogue. The dialogues contain common coordination phenomena, but the dataset task is to generate instructions, not to interpret them.

Ultimately, while the importance of communicative grounding and coordination is well-established, the specific mechanisms for handling miscommunications and repairs within situated, multi-modal dialogues remain relatively under-explored. Current research often bypasses these complexities, focusing on scenarios with simplified ambi-

guity or restricted interactivity. This overlooks a fundamental aspect of human collaboration: the ability to dynamically detect and resolve misunderstandings through interactive repair sequences. Addressing this gap is crucial for developing conversational agents that can robustly navigate the inherent uncertainties of real-world interactions. The following section outlines how this thesis tackles these challenges through specific research questions aimed at understanding and enabling agents to effectively manage miscommunications in multi-modal task-oriented dialogues.

## 2.4 Discussion

In this chapter, we have established the foundations of human communication, highlighting the crucial roles of communicative and symbol grounding (Section 2.1) and the repair mechanisms (Section 2.1.4) humans use to maintain mutual understanding. However, despite the critical importance of repairs in facilitating dialogue cooperation, few studies examine miscommunications as an integral part of tasks within multi-modal, situated environments.

As discussed in Section 2.3.4, current approaches and benchmarks often simplify or overlook the complexities of real-world collaboration, particularly the dynamic negotiation required to handle underspecified language and resolve miscommunications. This focus on idealized scenarios, often lacking realistic ambiguity or rich interactive phenomena, hinders the development of agents that can process miscommunications in situated human-agent dialogues.

To bridge this gap and enable agents to collaborate effectively with humans in shared environments, they must develop the capability to learn from and recover from communicative breakdowns. Conversational agents can leverage modern neural network architectures (reviewed in Section 2.2) to learn representations and ground the context (perception, dialogue history, etc.) and approach these challenges. The research questions explored in this thesis, discussed below, target key aspects necessary to achieve this goal, investigating how to elicit relevant phenomena and equip models with the representations and learning mechanisms needed to process miscommunications in multi-modal, task-oriented dialogues.

### 2.4.1 RQ1: What are the ways that we can elicit miscommunications in multi-modal referential conversations?

A recurring challenge in the field and highlighted in Section 2.3.4 is the lack of datasets capturing the nuances of collaborative dialogue, particularly requiring genuine cooperation to repair the miscommunications in situated, task-oriented settings. Many existing benchmarks (e.g., (Das et al., 2017; Padmakumar et al., 2022; Gao et al., 2022)) often rely on constrained data collection protocols (e.g., asynchronous interactions) or simplified scenarios (Suglia et al., 2024). This results in a lack of dialogues with context-dependent turns (Agarwal et al., 2020) or featuring ambiguities not easily resolved by more capable models, as they tend to fail to capture the phenomena required for communicative grounding. Consequently, these datasets often do not represent the dynamic negotiation of human collaboration.

This research question directly addresses this gap by investigating methods to elicit naturally occurring, complex miscommunications that require genuine cooperation and multi-modal grounding to resolve. The aim, explored in Chapter 3, is to collect realistic collaborative data within an instruction understanding task that is challenging to solve without human-agent cooperation.

### 2.4.2 RQ2: How can models learn robust multi-modal representations to resolve ambiguous referring expressions?

Resolving referring expressions (REC) is the first step towards learning to process their repairs in situated dialogue, jointly addressing both symbol and communicative grounding. As discussed, language is inherently underspecified, and in multi-modal contexts, expressions can be ambiguous, relying on a combination of visual context, dialogue history and shared knowledge to interpret them. Furthermore, when an interlocutor repairs an ambiguity, they typically provide just enough new information to help the addressee identify the referent, without repeating the entire instruction. Thus, processing repairs successfully requires having first built a reasonably accurate and adaptable representation of the initial ambiguous utterance.

This research question explores how VLMs, which integrate visual and linguistic information, can develop the necessary representations for REC in these challenging sce-

narios. Specifically, Chapter 4 investigates the capabilities of these models to both effectively ground linguistic terms to visual referents, addressing the symbol grounding problem, and integrate dialogue context across turns to resolve subsequent, potentially underspecified expressions. To handle the ambiguity inherent in situated dialogue, agents built with multi-modal neural architectures (Section 2.2) must develop robust representations for addressing the symbol grounding problem and thus REC. We evaluate how effectively these models can leverage multi-modal context not only to identify the correct referent but also potentially detect when an expression is ambiguous, a crucial first step towards processing miscommunications in situated dialogues.

### *2.4.3 RQ3: What are the signals necessary for learning to process repairs?*

While RQ2 focuses on building the initial representations needed to understand potentially ambiguous references, this question addresses how models can process corrective feedback or repairs. As established in Section 2.1.4, repairs are vital for cooperative dialogue, allowing interlocutors to resolve misunderstandings that arise in dialogue. In situated interactions, these repairs are often multi-modal, incorporating linguistic cues alongside non-linguistic information (Benotti and Blackburn, 2021), such as perception.

True collaboration requires handling communicative breakdowns with repairs, but it is unclear what model characteristics are required for learning this. Therefore, RQ3 investigates the specific signals and learning conditions required for models to successfully process repairs, particularly focusing on CEs (Section 2.1.4). Chapter 5 examines how different model architectures and training regimes (informed by the deep learning approaches discussed in Section 2.2) affect a model's ability to integrate new information from the CEs. To facilitate this analysis, we introduce a taxonomy classifying CEs based on the modalities required for their resolution, offering a more fine-grained classification than existing frameworks (introduced in Table 2.1 and Table 2.3) for understanding the multi-modal signals involved. The goal is to identify which architectural choices, training objectives and types of multi-modal input are most effective when learning representations to handle repairs in situated dialogue.

### 2.4.4 RQ4: How can we distil in models the capability to process repairs?

Successfully resolving referring expressions (RQ2) and identifying the necessary signals for learning to process repairs (RQ3) are crucial prerequisites, but they do not guarantee that models can effectively process repairs. As established in Section 2.1.4, repairs are complex, context-dependent phenomena (Purver, 2004; Purver and Ginzburg, 2004) essential for managing the inherent underspecification and potential ambiguities of natural language (Pezzelle, 2023; Ferreira, 2008) and maintaining communicative grounding (Clark, 1996; Mills, 2007). However, as reviewed in Section 2.3.4, current approaches in situated dialogue and instruction following often simplify or circumvent these collaborative challenges, limiting models' ability to handle the dynamic negotiation of meaning inherent in human interaction.

Therefore, this research question investigates specific training methodologies, leveraging advanced training techniques discussed in Section 2.2 (e.g., masking, RLHF), to equip models with the mechanisms needed to process and integrate repair information effectively. The goal is to move beyond simple ambiguity resolution and towards models that can adapt their predictions based on corrective feedback, similar to how humans repair miscommunications. To evaluate progress towards this goal, Chapter 6 introduces a benchmark specifically targeting Third Position Repairs (P3SISR), and we experiment with custom training strategies informed by insights from previous research questions. This addresses the gap concerning the lack of robust repair handling in current multimodal agents for situated dialogues.

# Chapter 3

# Crowd-Sourcing Repairs for a Collaborative Human-Robot Task

In this chapter, we present a preliminary study of human-robot interaction in collaborative settings. Dialogue cooperation requires agents to be aware of a wide range of contextual phenomena, including dialogue history, environmental surroundings, and the agent's own position and actions. To investigate the integration of multiple modalities beyond language and vision, we introduce a crowd-sourced dialogue corpus in a table-top manipulation task, where an agent actively participates and modifies the environment as the dialogue unfolds. The setting we choose in particular is highly ambiguous, eliciting miscommunications that both human and agent must iteratively resolve together. The resulting corpus, rich in context-dependent repairs (e.g., influenced by the robot's position or actions), provides an ideal foundation for researching miscommunication in situated, goal-oriented dialogues. We also report on exploratory modelling experiments with instruction-following agents in these scenarios to understand their capabilities and limitations.

## 3.1 Introduction

Effective collaboration depends on clear communication between dialogue participants, yet achieving perfect understanding is often challenging. Speakers produce utterances spontaneously, and many factors can lead to misinterpretations. For instance, speakers might produce malformed utterances, or listeners may mishear a word, altering the intended meaning. This problem is particularly pronounced in dynamic environments or when participants have mismatched perspectives (Dobnik et al., 2015), increasing the likelihood of miscommunications. Humans can then employ repairs to resolve misun-

derstandings, realign perspectives and continue to coordinate their verbal and non-verbal actions (Mills, 2014; Eshghi et al., 2022).

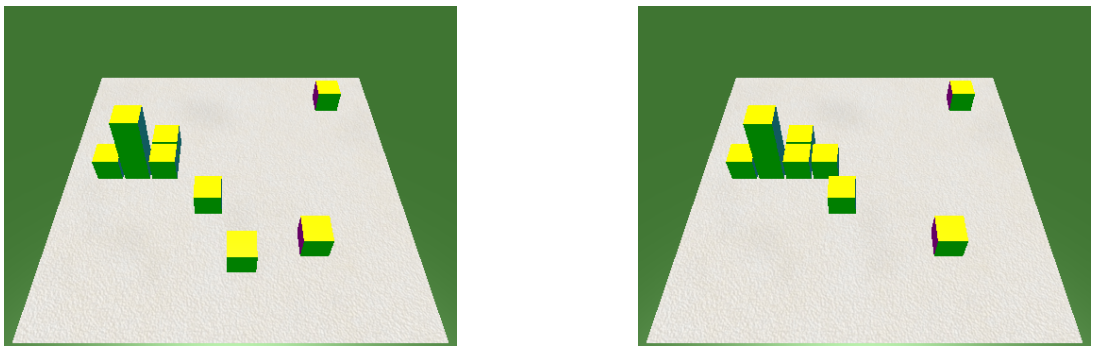
In contrast, NLP research in instruction understanding typically assumes that utterances are fully formed, unambiguous, and with clearly specified goals, focusing on well-formed single-turn instructions (e.g., *"Pick up the white mug on top of the kitchen table"*). Human language is however frequently underspecified (Pezzelle, 2023), lacking details to fully comprehend instructions, and instead relies on common ground between the conversation participants. For example, *"Pick up the mug on top of the table"* assumes that there is only one table, without mentioning its location or the colour of the mug.

Humans deal with underspecification in dialogues by inferring information from their surroundings, personal experiences or overall context (Grice, 1969), which is cognitively cheaper than producing fully specified utterances (Levinson, 2000). Integrating modalities and non-linguistic information is thus crucial for models to effectively collaborate with humans through language (Bender and Koller, 2020; Bisk et al., 2020) and deal with symbol grounding (Harnad, 1990).

This chapter approaches the underexplored challenge of ambiguous and underspecified natural language instructions in collaborative dialogues. We begin by collecting a dataset of situated dialogues within a human-robot instruction-following task set in a highly ambiguous environment, the Blocks World (Bisk et al., 2016) (Section 3.3). In this task, an agent has to identify a specific block to move and determine the target position based on natural language instructions (see Figure 3.1 for an example). However, since all the blocks appear identical, human instructions mix convoluted referring expressions grounded on other blocks or the world, which models have previously struggled to handle (see Tan and Bansal, 2018; Mehta and Goldwasser, 2019; Dan et al., 2021).

The task's complexity stems from the challenging referring expressions, which can be prone to misinterpretations and ambiguities (i.e., determining where an instruction describing a block's movement begins and ends relative to its final location). As shown in Figure 3.1, even humans may struggle with these nuances (also see $t_8$ of Figure 3.2).

Most research on instruction understanding focuses on models that continually improve their text comprehension, object recognition and reasoning capabilities, often assuming that humans provide clear and comprehensive instructions. However, this overlooks the inherent messiness and incompleteness of human language, where the meaning of (possibly succinct) utterances is frequently negotiated over several dialogue turns.



**Instruction:** *"Take the third block from the right and put it just to the right of the block that has the stack directly to its left."*

Figure 3.1: An example instruction from the Blocks World (Bisk et al., 2016), mixing which block to pick and where to place it. The instruction is convoluted and contains relative references to other blocks (e.g., *"...the block that has the stack..."*) or abstract concepts (e.g., 'stack'), which may lead to misinterpretations and ambiguities. As humans, we can resolve this instruction to a particular entity and action, but in this case, there is only one possible Frame of Reference.

Our approach differs by acknowledging the need for models to process incomplete or ambiguous instructions, including instructions with subsequent repairs or corrections, which are common in human-human collaboration. We argue that training on simple text instructions is insufficient for addressing the complexities of this task without incorporating dialogue cooperation, and that models must be able to handle conversational phenomena, such as incomplete or ambiguous utterances leading misunderstandings.

This variant of the Blocks World dataset uniquely combines challenges that require cooperation: the inherently ambiguous nature of its environment and the complexities of the human language in its instructions. Unlike other datasets that focus on individual objects within a scene, such as GuessWhat?! (de Vries et al., 2017), which heavily relies on visual cues like color or shape to identify objects, the Blocks World introduces ambiguity that cannot be resolved with i.e., better object recognition alone. This environment contains instructions that are inherently prone to misinterpretations, serving as our testbed for exploring human-agent dialogue cooperation. By leveraging the Blocks World, we can create scenarios where cooperation is inevitably required to overcome them, rather than models with superior individual capabilities (e.g., enhanced object detection or spatial understanding).

In this chapter, we investigate how human-agent conversations evolve in such ambiguous, situated contexts by crowd-sourcing dialogues using Amazon Mechanical Turk

(Section 3.4) set in the Blocks World. These dialogues, full of miscommunications and subsequent repairs, offer an ideal setting for learning to process imperfect instructions. We then train several instruction understanding models, aiming to improve their flexibility and resilience when faced with ambiguity and miscommunications (Section 3.5). Our results show that current model architectures struggle to integrate the multi-modality of language and to function effectively in these collaborative settings.

## 3.2 Related Works in Instruction Understanding

Early work on instruction understanding mapped language instructions to intermediate symbolic representations that are easily interpretable and executable (Chen and Mooney, 2011; Tellex et al., 2011; Artzi and Zettlemoyer, 2013). However, these approaches have largely been superseded by data-driven models or hybrid alternatives (Williams et al., 2017; Min et al., 2022) due to difficulties scaling handcrafted domain-specific modules (Blukis et al., 2018). More recent research has proposed models that combine vision and language, trained using deep learning (Bisk et al., 2016; Mei et al., 2016; Tan and Bansal, 2018; Suhr et al., 2019) or Reinforcement Learning (RL) (Misra et al., 2017; Blukis et al., 2018; Misra et al., 2018; Suhr and Artzi, 2018). Many of these works rely on simulated environments to assess models' generalisability, such as CHALET/LANI (Yan et al., 2018; Misra et al., 2018) or House3D (Wu et al., 2018), and broadly approach instruction understanding by encoding commands with an LSTM (Hochreiter and Schmidhuber, 1997) and visual observations with a CNN (LeCun et al., 1989), combining them into a final prediction.

However, these works so far mainly explore single-turn instructions and do not account for incomplete or underspecified instructions, nor consider the collaborative nature of language communication. Even works that explore instruction-following in dialogues, such as Talk The Walk (de Vries et al., 2018) or TEACh (Padmakumar et al., 2022), often treat dialogues as sequential instructions or simple question-answer pairs, as discussed in Chapter 2. Few works investigate how to handle and recover from errors based on iterative human feedback (e.g., Hough and Schlangen, 2017; Misra et al., 2018; Suhr et al., 2019; McCallum et al., 2023), as we explore here. Gervits et al. (2021) resolves references iteratively by exploiting unknown properties and minimising entropy with symbolic-probabilistic modelling, but ultimately mirroring the GuessWhat setup (de

Vries et al., 2017) without directly addressing vision or dialogue coordination itself. Similar to our approach here, Appelgren (2022) explore situated feedback in an interactive task, leveraging semantics to interpret feedback and improve learning efficiency.

Many existing environments (see discussion in 2.3.4) focus on settings that assess models' abilities to detect specific objects or navigate unseen scenarios, challenges that can often be addressed by models with superior individual capabilities, such as larger pre-trained models or enhanced object detection (e.g., Shridhar et al., 2020). In contrast, we prioritise tasks that require true collaboration to succeed. For this reason, we use the highly ambiguous Blocks World environment, where models cannot rely solely on language patterns or improved visual processing, and instead, they must demonstrate strong contextual awareness of the situated scenario, dialogue and their actions.

## 3.3 The Blocks World

Originally proposed by Winograd (1972), Bisk et al. (2016)[1] released a dataset of instructions to explore human-robot communication in this world as a situated environment. The task requires arranging blocks in a certain configuration according to crowd-sourced human instructions as a tabletop manipulation task, see Figure 3.2 for examples. These instructions are given in sequence, so the environment changes as the blocks get increasingly organised, exploring the idea of agents that can modify the world around them through language and not act solely as observers (Bisk et al., 2020). However, as shown in the example sequence in Figure 3.2, the instructions within these sequences are unrelated, so models do not need to ground their actions in the environment beyond identifying the block to move and where to move it.

The dataset comprises of three variants distinguished by the defining characteristics of the blocks: numbered (blocks labelled with numerical identifiers); logos (blocks with distinctive brand logos); and blank (blocks without distinguishing features). We focus on the later version of the dataset where blocks have no defining features, they are **blank**, and thus instructions must resort to challenging spatial referring expressions to locate blocks and positions. This contrasts other datasets or tasks in the field that reduce ambiguity to a minimum by using objects with clearly distinguishable visual properties, such as 'blue cube' vs 'red sphere' as in CLEVR (Johnson et al., 2017). These datasets can

[1] Later expanded in Bisk et al. (2018).

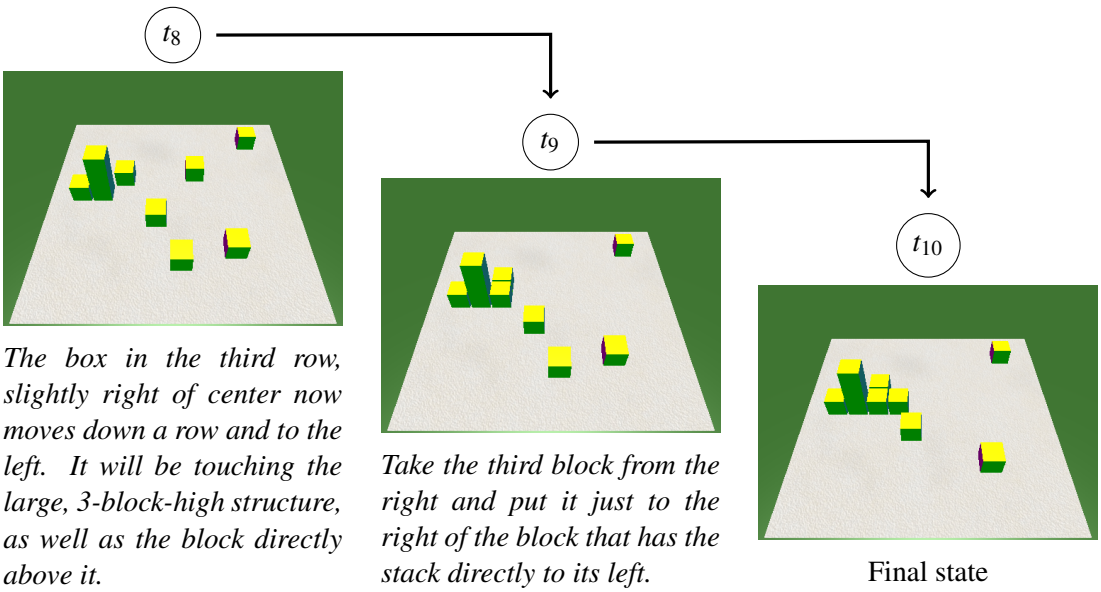



Figure 3.2: Sequence of states $t$ with corresponding instructions with blank blocks from the Blocks World (Bisk et al., 2016).

often be addressed with stronger visual representations alone (e.g., disentangled object representations (Bengio et al., 2013)), unlike the blank version of the Blocks World, which due to its high ambiguity, fine-grained reasoning and human-agent understanding plays a crucial role.

**Ambiguity** The environment introduces ambiguity across multiple dimensions due to the uniform appearance of all the scene blocks, paired with extreme information asymmetry between the human and the agent (humans posses all relevant knowledge). The spatial referring expressions in instructions are typically under-specified, exacerbated by a lack of differentiable points of reference (i.e., a landmark in instruction-following tasks (Katsakioris et al., 2021)). This results in a large variability and thus lexical ambiguity in the instructions' language, as 'the block on the right' could refer to the block's absolute position on the table or its position relative to another object, and abstract concepts like 'third row' can be misinterpreted over whether it initially refers to the top-most or bottom-most row of blocks ($t_8$ in Figure 3.2). Furthermore, relative clauses introduce additional ambiguity, such as determining the meaning of 'the third block from the second column' or finding where an instruction describing a block begins and ends relative to its final location. The overall challenge lies on agents both interpreting spatial relationships and understanding the context-dependent nature of the instructions.

A limitation of the Blocks World dataset is its primary focus on ambiguity aris-

ing from spatial relations, overlooking other potential sources of ambiguity common in real-world interactions, such as visual or action ambiguity. Works exploring visual dialogues, such as GuessWhat?! (de Vries et al., 2017) or CLEVR-Dialog (Kottur et al., 2019), tend to simplify the ambiguity by ensuring target objects have unique combinations of characteristics. Thus, ambiguities can be trivially resolved with simple attribute-based questions and deterministic slot-filling (e.g., *"Is the object red?"* → `[colour=red,shape=square]`), sometimes circumventing the need for dialogue altogether (Agarwal et al., 2020). Increasing visual clutter in these datasets (e.g., by adding more objects with similar features) would heighten visual ambiguity, yet it might not fully capture the inherent underspecification of the spatial language seen in the Blocks World. For instance, the phrase *"The block on the right"* can be ambiguous, referring either to a block's absolute position on the table or its position relative to another object. It is this type of linguistic underspecification that requires clarification and negotiation, making it particularly relevant for studying natural human-agent communication.

### *3.3.1 Sub-Tasks*

We identify two sub-tasks in the Blocks World: (1) **source block prediction** - determining which block should be moved, often framed as a classification task over a set of blocks; and (2) **target position prediction** - determining where to move it as a table location in the 3D space, treated as a regression task. These tasks are primarily evaluated using block distance (Bisk et al., 2016), defined as the Euclidean distance between the predicted and true positions (XYZ coordinates), normalised by block length in the 3D space (lower values indicate better performance; see Equation (3.1)). Source block prediction can also be assessed with accuracy, where higher values are better.

$$\textit{Block Distance}(xyz_{pred}, xyz_{true}) = \frac{\sqrt{(x_{true}-x_{pred})^2+(y_{true}-y_{pred})^2+(z_{true}-z_{pred})^2}}{\textit{BlockLength}} \tag{3.1}$$

$$\text{Accuracy} = \frac{\textit{TruePositives}+\textit{TrueNegatives}}{\textit{TruePositives}+\textit{TrueNegatives}+\textit{FalsePositives}+\textit{FalseNegatives}}$$

### 3.3.2 Prior Modelling Approaches

Several works have used the Blocks World to investigate instruction understanding (Misra et al., 2017; Bisk et al., 2018; Platonov et al., 2020), tough only a few tackle the challenging blank-labelled version of the dataset. Bisk et al. (2016) initially proposed a baseline end-to-end RNN architecture that struggled with these instructions. Tan and Bansal (2018) developed a multi-modal vision-and-language model, similar in architecture to LXMERT (Tan and Bansal, 2019), leveraging dual attention and joint training for both sub-tasks, achieving the highest scores at the time of this writing (source block prediction accuracy of 56%, and 3.07 mean block distance from target location). To improve model robustness, Dan et al. (2021) introduced adversarial data augmentation techniques, revealing that existing instruction-following models were highly sensitive to perturbed instructions. Most relevant to our work, Mehta and Goldwasser (2019) relaxed the assumption that one natural instruction directly corresponds to one action. They proposed a language-only model that incorporates intermediate advice to refine its predictions. Although this advice aims to simulate human insight, it is automatically generated based on table regions (e.g., top-left) and the prediction still happens in a single turn.

## 3.4 Eliciting Human-Robot Cooperation in the Blocks World

Instructions in the Blocks World are often long and convoluted, incorporating referring expressions for both the block and its target location. For example:

> *"Take the top left block and put it so its near left corner nearly touches the far right corner of the stack of blocks"*

In these cases, even highly detailed expressions may fail to uniquely identify the intended referent. Mehta and Goldwasser (2019) already demonstrates that incorporating approximate advice can benefit models in single-turn instruction understanding. We build on this by arguing that interactive settings where agents engage in clarificational dialogues, akin to human-human collaboration, can mitigate misunderstandings or shortcomings in instruction understanding and improve models' robustness. Through dialogue, subsequent turns would introduce new disambiguating information, helping agents narrow down the correct block or location. This would enable agents to learn coordination through language in a multi-modal situated task, much like how humans resolve referential ambiguities in everyday conversations (e.g., *"Bring that mug* $\rightarrow$ *Which one?* $\rightarrow$ *The red one on the table"*). Cooperating through language and across modalities is a hallmark of truly collaborative agents (Suglia et al., 2024; Schlangen, 2023).

Unlike more recent photo-realistic simulated environments such as AI2Thor (Kolve et al., 2017) or ALFRED (Shridhar et al., 2020), the Blocks World intentionally sacrifices vision complexity to focus on assessing models' core instruction-understanding capabilities in an ambiguous robotic manipulation task. This approach is similar to VIMA-Bench (Jiang et al., 2023b), which tests models' generalisability by simulating novel object configurations within a visually-simple yet restricted space. However, VIMA-Bench primarily introduces ambiguity through a lack of knowledge about the object mentioned (e.g., *"What is a wug?"*), similar to underspecified tasks in ALFRED (Shridhar et al., 2020), TEACh (Padmakumar et al., 2022) or DialFRED (Gao et al., 2022) (i.e., *"Make me a coffee → Where is the coffee mug?"*).

Few works on instruction understanding consider dialogue as a way to resolve ambiguities (Padmakumar et al., 2022; Gao et al., 2022), but in most of these cases the ambiguity is minimised so that strong visual representations are enough to resolve them (Gao et al., 2023b) (e.g., *"yellow or blue spoon?"*) or do not involve a situated 3D environment where the agent must both reason and act (Dobnik et al., 2020; Madureira and Schlangen, 2023). These approaches also rarely feature true bi-directional information flow (*human* $\leftrightarrows$ *agent*), typically assuming that a human oracle provides clear and complete instructions for the agent to follow (*human* $\rightarrow$ *agent*). Language games investigate this notion of bi-directional information flow, but popular benchmarks are also limited by a lack of coordination phenomena (de Vries et al., 2017) or simple question-answer pairs for referential ambiguities (Haber et al., 2019). This neglects the spontaneity of dialogue (e.g., malformed utterances) and overlooks the cooperative nature of communication, where both parties actively collaborate in the interaction (Schlangen, 2023).

Thus, to explore human-agent communication in ambiguous, situated environments, we crowd-source dialogues through Amazon Mechanical Turk (AMTurk) and extend the Blocks World with dialogues situated in a 3D environment. We run this in a simulation, which allows a robotic arm agent to move blocks and modify the environment itself, a first step towards collaborative agents (Bisk et al., 2020). This additionally enables the agent to be an active participant in the dialogues, pointing to objects, executing incorrect actions, and humans to correct their own or the agent's misunderstandings, similar to previous works (Hough and Schlangen, 2016; Hough and Schlangen, 2017). The rest of this section explains the simulated robot environment, the data collection methodology, manual annotations and finally provides a corpus analysis. We will use this data in the next section to train models for collaborative situated tasks.

### 3.4.1 Simulated Robot Environment

A key indicator of true collaboration is an agent that can both act and reason about the consequences of its actions within its surrounding environment (Bisk et al., 2020). While the original Blocks World provides images, natural language instructions and block coordinates in 3D space, it lacks an actual simulation environment. Some later efforts simulated simplified settings, such as top-down views (e.g., Misra et al., 2017), but these approaches typically overlook the agent's role as an active participant in the world.

To address this, we developed a simulated environment that emulates the Blocks World in Gazebo Sim, a simulation platform well-suited for robotics tasks with a realistic physics engine[2]. We integrate a robotic arm[3] equipped with a vacuum gripper to manipulate blocks and controllable through an API based on ROS[4]. The robot signals its intent by pointing the gripper and lighting up the target block or position inspired by our previous research in explainable robotics (Chiyah Garcia et al., 2018b; Chiyah Garcia et al., 2020a; Chiyah Garcia et al., 2021). We use the block coordinates from the Blocks World to position the objects on the table, Figure 3.3 shows the final simulation. The robot arm introduces additional considerations to the task beyond embodiment, such as the potential for hardware failures (e.g., failing to pick up a block) and the need to account for movements over time (e.g., moving to an incorrect point or picking up the wrong block). These challenges highlight the importance of real-time interventions and repairs for resilient human-robot communication, elements that static text alone cannot capture and that are often neglected in instruction understanding tasks, yet are crucial for effective, situated collaborative AI.

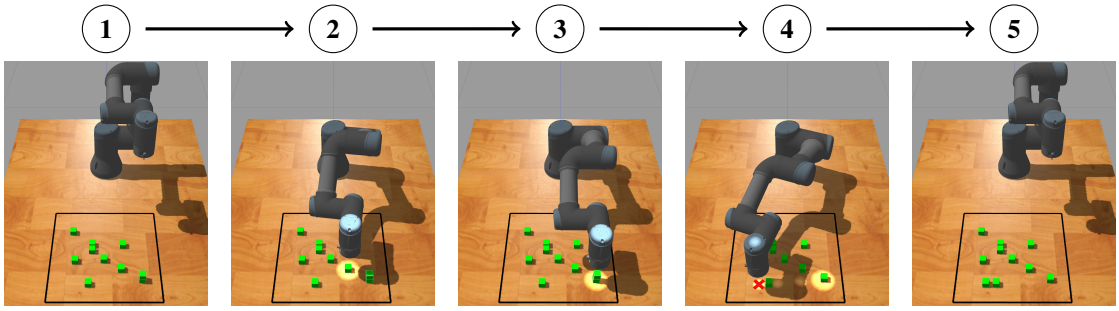



Figure 3.3: Dialogue in our simulated Blocks World environment with the embodied robot arm.

[2]See simulation platform at `https://gazebosim.org/`
[3]Based on a Univeral Robot 3, `https://www.universal-robots.com/products/ur3-robot/`
[4]Robot Operating System, a common tool package for robotics applications, `https://www.ros.org`

### 3.4.2 Data Collection

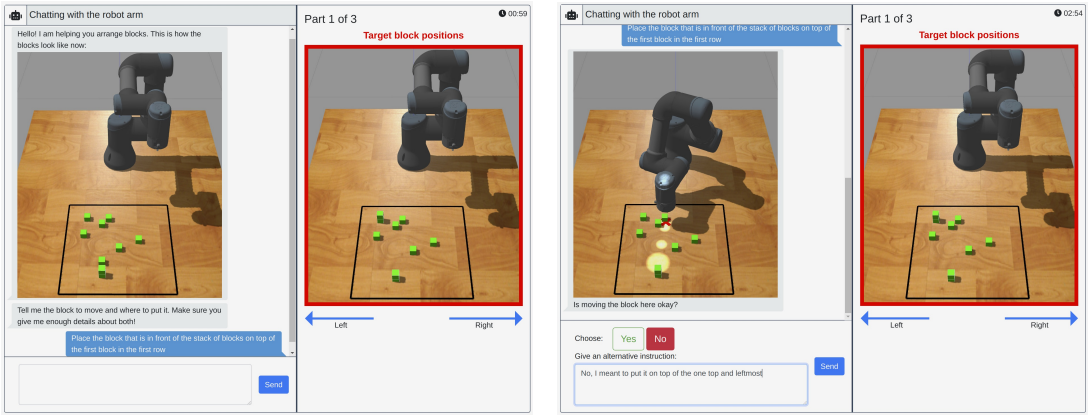
Workers in AMTurk interacted with a rule-based dialogue system emulating an agent moving blocks in the simulated environment. The workers' task was to provide instructions to move blocks according to a reference image, similar to the original Blocks World dataset collection. However, we designed the robot agent to occasionally make intentional errors and to verify its actions by pointing to candidate blocks or locations. AMTurk workers can then collaborate with the robot to correct misunderstandings via short dialogues based on the robot's indicated position.

The robot performs the correct action 70% of the time[5] and, after a few turns of back-and-forth repairs (maximum of 2 turns) where the robot attempts to understand the user, it will select the appropriate block or location, ending the dialogue. Only one of the actions would be disambiguated at a time, with the robot first confirming the block to move (source block), then picking it up, and finally disambiguating the location (target position) if needed. The implemented randomness in dialogues ensured that workers were in a variety of situations over the course of the 4 dialogues they would complete, i.e., in one dialogue, the robot would fully resolve the instruction after 1 turn, whereas other times the robot would need the maximum number of disambiguation turns.

Workers communicated with the agent via a chat interface (see Figure 3.4) and could type messages freely, resulting in mostly unstructured interactions. The agent would send images of the simulated world in every reply, allowing workers to track how their dialogue changed the table state over time (e.g., seeing where the agent pointed or when it held a block). To manage the data collection and world states, we used the CRWIZ framework (Chiyah Garcia et al., 2020b). Originally built upon Slurk (Schlangen et al., 2018), we developed this framework to facilitate data collections pairing multiple AMTurk workers in chat-based simulated scenarios, as in our previous work (see Lopes et al., 2020). In this data collection, where workers did not interact with each other, we used CRWIZ to manage many of the back-end details, including the robot's verbal and non-verbal actions, and effectively running a simulated sequence of events, often unique for each AMTurk worker.

At the end of the task, workers completed a questionnaire that included the Perceived Intelligence scale from the Godspeed Questionnaire (Bartneck et al., 2009) to ensure that the task was not too difficult and the agent was perceived as intelligent (despite its rule-based dialogue and continuous misunderstandings). We provide an example of the AMTurk interface in Figure 3.4.

[5] As previously mentioned, the best models for this setting of the Blocks World only pick the correct source block 56% of the time (Tan and Bansal, 2018).




(a) User provides an initial instruction with both the source block and target position.

(b) User provides a correction after incorrect robot action due to a misunderstanding.

Figure 3.4: AMTurk interface for the data collection.

### 3.4.3 Collected Corpus

We collect a total of 796 dialogues, after filtering out problematic ones from AMTurk, with 4,325 user turns, see Dialogue 9 for an example. Each dialogue is automatically annotated with rich metadata, including block coordinates, robot positions, executed actions, and user utterances. Following common practice in robotics, we used a frame-based approach, recording metadata at every time step, allowing for an exact reproduction of events. We carry out a manual corpus annotation and then provide a detailed analysis in Section 3.4.5.

*(9)* Sample dialogue from the corpus.

**Agent:** *Hello! I am helping you arrange blocks. This is how the blocks look like now:* `<show image>`

**Agent:** *Which block should I move and where to? Make sure you tell me both*

**User:** *of the three blocks closest to you, move the block on the right to the right of the stack of blocks*

**Agent:** `<show image>` *Is it this block?*

**User:** *the one just adjacent right of that*

**Agent:** `<show image>` *Do you mean this block?*

**User:** *No one much more closer to the previous block you selected*

**Agent:** `<show image>` *Can you confirm it is this block?*

**User:** *Yes*

### 3.4.4 Manual Annotations

The corpus dialogues contain many idiosyncrasies that difficult separately parsing which section of the utterance refers to the source block and which indicates the target position. Users referred to blocks and locations in highly varied and unconstrained ways, often using different frames of reference or even abstract structures, similar to that reported by Bisk et al. (2018) (e.g., *"...between blocks in a line in the last row."*).

To ensure data quality, we manually reviewed the dialogues, flagging or filtering out any problematic instances. Additionally, we compiled a list of keywords that humans commonly used to describe blocks and locations (e.g., 'topmost', 'northwest'). We then applied regular expressions to parse and annotate the dataset based on these keywords automatically.

### 3.4.5 Corpus Analysis

| **Measure** | **#** |
|---|---|
| Dialogues collected | 796 |
| Dialogues with 0 corrections | 28 |
| Dialogues with 1 correction | 174 |
| Dialogues with 2+ corrections | 535 |
| User all turns | 4,363 |
| User instruction turns | 796 |
| User correction turns | 2,282 |
| User acknowledgement turns (e.g., *"Yes"*) | 1,286 |
| User total tokens | 37,144 |
| User unique tokens | 752 |
| Mean tokens per user utterance (SD) | 11.18 (6.52) |
| Mean user turns per dialogue (SD) | 3.21 (1.54) |
| Mean user corrections per dialogue (SD) | 2.21 (1.54) |
| User unique bigrams | 4,257 |
| User unique trigrams | 9,024 |
| User utterances referencing abstract structures (e.g., row, column) | 765 (17.5%) |

Table 3.1: General statistics of the collected dialogues.

Table 3.1 provides a summary of the corpus statistics. We differentiate user turns into three main types: 1) the initial user instructions (one per dialogue); 2) user corrections, which address either the source block or target position (averaging 2.21 per dialogue); and 3) acknowledgements, representing positive feedback to confirm a selection and continue the dialogue (e.g., *"Yes"*, with

a maximum of 2 per dialogue). We find that most dialogues include at least two corrections, with an average of 3.21 user turns per dialogue. Additionally, we observe that user utterances combine absolute references (e.g., *"... the topmost block"*) with relative ones (e.g., *"... the block above that one"*), and often reference abstract 3D concepts (e.g., row, column), which may be challenging for models to interpret (Bisk et al., 2018).

For further analysis, we use NLTK[6] (Bird et al., 2009) to split the user utterances into tokens (words). The utterances are widely diverse, containing 752 unique tokens, with many more unique bigrams and trigrams. We also computed several measures of lexical richness and compare them with the original Blocks World instructions in Table 3.2. Despite the similar overall size, the original instructions are significantly longer and contain more tokens, which is expected as corrections focused on smaller adjustments (e.g., *"Move left slightly"*).

To assess lexical diversity (Torruella and Capsada, 2013), we measure vocabulary variation within the text using the Type-Token Ratio (TTR) and the Mean Segmental Type-Token Ratio (MSTTR) (Lu, 2012). According to TTR, our corpus shows slightly higher diversity than the original Blocks World instructions; however, this trend reverses with MSTTR, a more reliable metric that adjusts for text length by averaging over segments. For context, the E2E corpus (Novikova et al., 2017), which covers real-world data in the restaurant domain, achieves an MSTTR of 0.75. In general, this suggests that both the Blocks World and our corpus maintain comparable levels of lexical diversity, despite focusing on a single task and narrow domain.

| **Measure** | **Our Corpus** | **Blocks World** |
|---|---|---|
| Utterances (instructions/corrections) | 3,078 | 3,573 |
| User total tokens | 37,144 | 78,791 |
| User unique tokens | 752 | 1078 |
| Mean tokens per utterance (SD) | 11.18 (6.52) | 22.05 (8.78) |
| Type-Token Ratio | 0.017 | 0.014 |
| Mean Segmental Type-Token Ratio | 0.643 | 0.713 |

Table 3.2: Comparison between collected corpus and original Blocks World instructions.

## 3.5 Modelling Situated Interaction

In this section, we conduct an exploratory study of model capabilities for handling miscommunications in our collected dialogue corpus. We examine methods for grounding language and actions in the world state, aiming to learn collaboration strategies for interactive settings. To achieve instruction understanding in these Blocks World dialogues, models must grasp the broader context

[6]A toolkit to analyse natural language with Python, see `https://www.nltk.org/`

(e.g., dialogue history, actions performed, spatial relationships), and crucially, must adapt their incorrect actions in response to user repairs. Through a series of experiments with different model architectures, we assess models' capabilities to handle these dialogues and correctly interpret user repairs. We begin with a model based on HCN, well-suited for environments where the robot and human are co-located, followed by a model that leverages simple vision to process world states. Finally, we conclude with a case study of a simplified RL agent designed to learn how to interpret and adjust to repairs.

### *3.5.1 Experimental Setup*

We formalise the experimental setup for the model experiments as follow.

#### Dataset

We use the collected corpus from Section 3.4. These dialogues contain between 3 and 7 user turns (including repairs and acknowledgements), which we segmented at each user turn to obtain 4,363 tuples of *(user turn, dialogue history, action history, world state)*. Each turn is paired with the robot's next action in the simulation, along with the corresponding true predictions from the Blocks World (source block and target position), to define our prediction goals: *(next action, source block, target position coordinates)*. We split the data into 70/10/20 for train/validation/test sets, respectively, and evaluate on unseen world configurations.

#### Task

Our goal is to develop models that can not only follow instructions but also act and interact effectively within the environment. We therefore expect models to predict their next action and identify both the block to move and its target position. The two sub-tasks in the Blocks World share common features (e.g., initial language instruction, world state) and could be combined as a unified task of predicting coordinates (whether for a block or a location).

We focus on these two main objectives:

(1) **Action prediction:** predicting the next action, either linguistic ($A_{dialogue}$) or non-linguistic ($A_{physical}$). By combining physical movements with dialogue intents, we aim to learn a

holistic interactive behaviour for predicting $A$.

$$A_{dialogue} = \{greet, request\_command, clarify\_source, clarify\_target, end\}$$
$$A_{physical} = \{move, point, pickup, dropoff\}$$
$$A = A_{dialogue} + A_{physical}$$

(2) **Coordinate prediction:** predicting the board coordinates corresponding to a referenced object or location. We focus on the $XZ$ plane (omitting $Y$-axis height) as $Y$ is always the same unless blocks are stacked, and our simulated robot can only interact with top-facing blocks. We match these coordinates to those of the source block or target position when relevant, and predictions are evaluated only for actions that require spatial information, $\{$*move, clarify_source, clarify_target*$\}$ (e.g., *move* requires coordinates, while $pickup$ and $dropoff$ operate directly under the robot gripper's position).

### Metrics

To evaluate action prediction as a classification task, we use **accuracy** against the next robot action. For coordinate prediction, we use the same metrics as the Blocks World depending on the relevant sub-task, **block distances**, refer to Section 3.3.1 further details. Each of our dialogue segments only focus on one of the sub-tasks at a time, so we can evaluate these separately and then present the aggregated results.

## *3.5.2 Models*

We conduct experiments with two different models.

### Hybrid Code Networks

Interactions co-located with humans in robotics have strict safety requirements that data-driven end-to-end models cannot guarantee. For instance, a robot collaborating with a human may predict actions that could pose a safety risk if the prediction is *move* in close proximity to a human operator. HCN (Williams et al., 2017) address this by combining data-driven techniques with domain-specific knowledge to manage dialogue flow and interactions. HCNs are highly efficient in learning from small datasets, reportedly between 100 to 200 dialogues in the original work. We also show in Lopes et al. (2020) that an HCN model can outperform off-the-shelf pre-trained models such as BERT (Devlin et al., 2019) with just around 200 dialogues. The integration of domain rules also enhances safety, enabling their deployment in close proximity to humans for

situated collaboration.

Building upon these HCN works (Williams et al., 2017; Lopes et al., 2020), we experiment with a HCN for handling dialogues in this environment. The action mask makes the model's output inherently safe, avoiding any uncontrolled or unexpected actions in situated scenarios, and we can still train it to learn our desired behaviour. This model is also highly suitable in settings where previous history is relevant, as in our case with prior dialogue or actions. We expect that this HCN quickly learns to encode the language and conversation flow used in the dialogues.

**Input** We use the following input features for our HCN (shown in Figure 3.5): (1) a Bag-of-Words (BoW) representation of the user utterance and dialogue history so far; (2) a one-hot vector encoding of the previous actions; (3) an action mask indicating currently allowed actions; (4) word embeddings extracted with the method proposed by Williams et al. (2017) and then averaged with word2vec embeddings (Mikolov et al., 2013); and (5) a context vector representing the world state to ground actions to the environment. The context vector consists of block coordinates for each block (3 float values for 10 blocks), the current's robot state (XY position coordinates, a binary value denoting if a block is attached) and relative position features proposed by Tan and Bansal (2018) for each block (Euclidean distances to board corners and edges, 8 floats, and a binary value denoting stacked blocks).

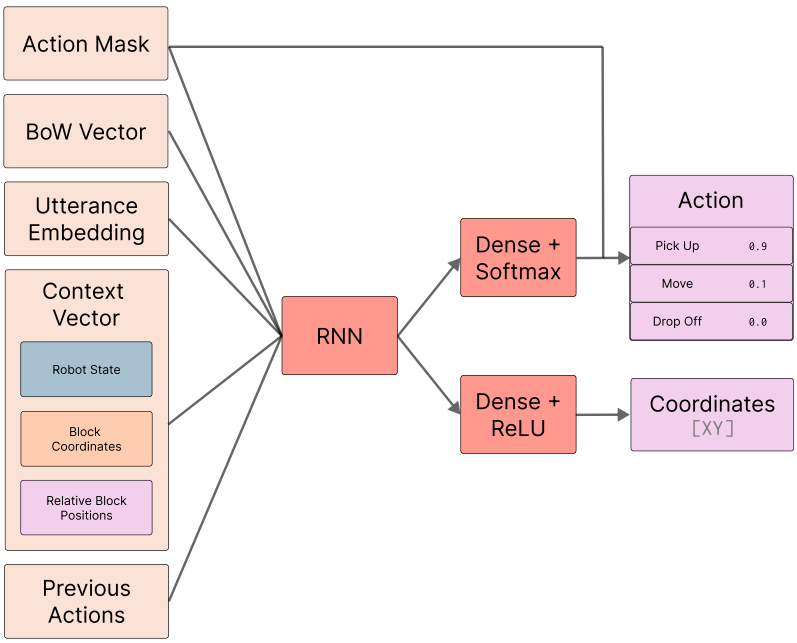



Figure 3.5: Architecture of the HCN with the input features and output results.

**Output** The input features are flattened and concatenated into a single vector. We then pass it to an set of RNN layers, which computes a hidden state. This state goes through both 1) a dense

layer with Softmax activation to obtain a probability distribution over the possible actions; and 2) a dense layer with ReLU (Dahl et al., 2013) activation normalised to obtain the $XYZ$ coordinates[7] in the range 0-1, which we project to the 3D environment.

**Training** We use LSTMs (Hochreiter and Schmidhuber, 1997) for the recurrent layer and set the number of hidden units to 128 following (Williams et al., 2017). We train for 12 epochs using categorical cross-entropy loss on the action output and mean squared error loss for the coordinates, as is common in regression tasks. We also use the Adam optimiser (Kingma and Ba, 2015) and $1e{-}3$ as learning rate.

### CNN-Based Model

An alternative approach for grounding language in the world is to encode the state with a CNN (LeCun et al., 1989), allowing the model to learn and focus on relevant areas. This technique has been used in prior work for the Blocks World, such as by Misra et al. (2017), which uses a CNN to process the visual state with non-blank blocks, and is typically combined with the language encoding to enable joint reasoning across modalities.

Our coordinate prediction task also presents additional challenges for optimisation, as coordinates exist in a continuous space that does not directly align with the discrete visual observations of a CNN. To aid this prediction, we discretise the output space into regions, following prior approaches in instruction understanding that reformulate the task as calculating a probability distribution over image pixels (Misra et al., 2018) or over cardinal actions for block placements (e.g., using 81 discrete actions as in Misra et al., 2017). By discretising the output space, we can effectively optimise both action prediction and coordinate prediction as classification problems.

Therefore, we propose a model that combines a CNN for encoding the world state and an RNN to process language, enabling it to predict the robot's next action and a position as a region on the board. In this case, the CNN should learn to encode the perceived world state, representing key perceptual features such as blocks and their spatial relationships within the world. The CNN's learned visual representation can then be integrated with the encoded language to predict positions and adjust them with the corrections accordingly, grounding the language in the world state. We set the board size to 10 by 10 regions, which provides enough fine-grained details whilst constraining the state space[8]. Any board discretisation introduces a degree of noise, as positions cannot be represented with complete accuracy.

[7] We match the format of the input block coordinates to the output, but we omit $Z$-coordinates from evaluation.

[8] Each block measures approximately 0.15 units, so a 'perfect' discretisation would require a 14x14 – almost double the size of the 10x10 board.

**Input** We use the following input features for our model, also shown in Figure 3.6: (1) a text sequence for the user's current utterance; (2) a one-hot vector encoding of the previous action; (3) the previous candidate position, if available; and (4) an image-based representation of the world state as an automatically-generated top-down view vector indicating block locations. We design this model to handle human feedback explicitly by incorporating the candidate output, representing the location where the robot was pointing before picking up or placing a block. We expect that in such cases, the model can learn to interpret corrections like *"A little to the left"* and predict a region shifted left from the previous candidate position.

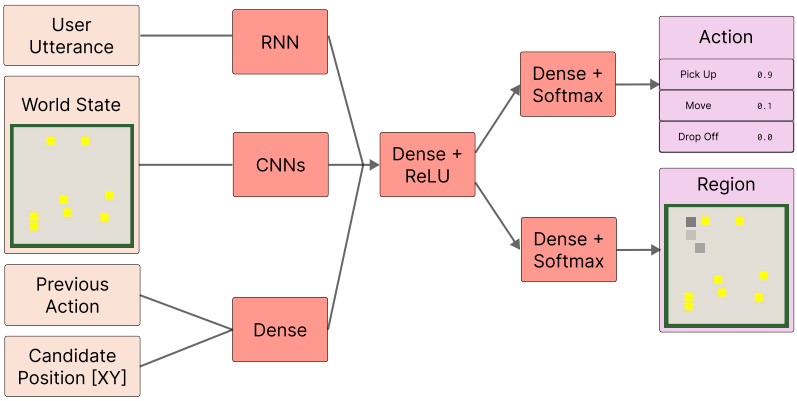



Figure 3.6: Architecture of the CNN-based model with the input features and output results.

**Output** The world state representation is encoded using a series of CNN layers with ReLU (Dahl et al., 2013) activations, flattened and passed through a dense layer to produce a hidden world state. The text sequence is encoded with an RNN to obtain a hidden language state. The previous action and candidate position are encoded separately through dense layers, and then concatenated with the hidden world and language states. This concatenated vector is passed through a dense layer with ReLU activations to form the final representation.

The outputs are generated from: 1) passing the final state through a dense layer with Softmax activation to produce a probability distribution over possible robot actions; and 2) using another dense layer with Softmax activation to predict a probability distribution over regions on the 10x10 board.

**Training** As in Section 3.5.2, we use LSTMs (Hochreiter and Schmidhuber, 1997) for the recurrent layer, setting the number of hidden units to 128, following (Williams et al., 2017). We train for 6 epochs as we do not see benefit from longer training. We also apply sparse categorical cross-entropy losses for both the action and position outputs and optimise using Adam (Kingma and Ba, 2015) with a learning rate of $1e{-}3$. For evaluation, we use the centre of the predicted

region to calculate block distances, and for source block accuracy, we check whether the correct block exists at that predicted location.

### 3.5.3 Results and Discussion

We present and analyse the results from training the models described above, comparing them against previous approaches on the blank-labelled version of Blocks World and baselines, as shown in Table 3.3. Unlike prior works that solely focus on source block and target position prediction, we approach this as a collaborative task of moving blocks while interacting a human. This means that our approaches are fundamentally different, as we consider actions and predictions over multiple steps.

| Model | Action | Source | | | Target | |
|---|---|---|---|---|---|---|
| | Acc % | Acc % | Median | Mean | Median | Mean |
| Random Baseline | 11.1 | 10.0 | 4.90 | 4.97 | 5.51 | 5.44 |
| Majority Baseline | 61.4 | 10.0 | - | - | - | - |
| **Our Models** | | | | | | |
| Hybrid-Code Network | 88.9 | 10.0 | 3.34 | 3.51 | 3.58 | 3.62 |
| Discretised World - CNN+LSTM | 91.2 | 10.1 | 3.30 | 3.48 | 3.58 | 3.61 |
| **Previous Approaches** | | | | | | |
| RNN (Bisk et al., 2016) | - | 10.0 | 3.29 | 3.47 | 3.60 | 3.70 |
| Best Ensemble (Tan and Bansal, 2018) | - | 56.8 | 0.00 | 2.21 | 2.78 | 3.07 |
| 4 Regions (Mehta and Goldwasser, 2019) | - | - | 2.23 | 2.21 | 2.18 | 2.19 |

Table 3.3: Comparison of our models trained on the corpus for jointly predicting the next robot's action and position of the source block/target location. We measure accuracy (Acc; higher is better ↑) as a percentage and median/mean as normalised block distances (lower is better ↓). The majority baseline uses the most frequent action from the transition probabilities at each turn of the dialogue.

**HCN** This model achieves strong results for predicting the next robot action (e.g., move, clarify), 88.9% accuracy, and outperforms baselines by a large margin. However, its performance in coordinate prediction is limited, with an average error of 3.5 blocks away from the correct source block or target position, comparable to the RNN model proposed by Bisk et al. (2016). The model struggles to incorporate corrective user feedback, often predicting coordinates that do not align with the correction's direction. While HCNs are typically effective at managing dialogue flow (Williams et al., 2017), we observe a tendency for this model to predict the end action prematurely, even if the user has not confirmed successful block placement in our test dialogue scenarios. This suggests that the model fails to effectively integrate references to blocks or positions from the visual context into its action predictions.

**CNN-Based Model** This approach achieves 91% accuracy in action prediction for the next action in our test dialogues, but struggles to identify the correct region. Given that the world state is now discretised into 100 fixed regions, we would expect that this task would be easier than predicting exact coordinates, which the HCN found particularly challenging. For instance, discretising actions allowed the model in Misra et al. (2017) to generate finer-grained movements compared to models that predict only final positions, such as (Bisk et al., 2016). However, despite learning to identify regions containing blocks for the source block prediction sub-task, we do not see any performance improvements compared to our last model, with no adjustments following user corrections. This indicates a lack of grounding between the model's actions and the world context.

The HCN and CNN-based models both exhibited sub-optimal performance due to their inability to ground the natural language and the visual context. The HCN struggled to link linguistic references to the scene for position predictions, suggesting that the world state input may have not been suitable, whereas the CNN, despite processing visual data, failed to leverage this context or user corrections to adjust target regions.

### *3.5.4 Learning to Follow Corrections with Reinforcement Learning*

In this section, we use Reinforcement Learning (RL) as a case study for learning to follow human corrections. While our previous models (Section 3.5.2) successfully learned to perform actions within the environment, they struggled with accurately predicting positions. A key advantage of enabling cooperation through dialogue is that even when human and robot perspectives are misaligned (as they often are (Kruijff, 2012)), feedback exchanges allow the human and agent to iteratively repair and ground the dialogue (Hough and Schlangen, 2016).

In our corpus, humans frequently refine robot actions through incremental corrections, adding new details and referencing prior actions or the robot's current position. These goal revisions (for source block or target position) should guide the model to adjust its previous predictions (e.g., if corrected with 'right', the new position should shift accordingly). However, our previous models trained with supervised learning showed limited improvement in processing corrections.

Here, we investigate this concept of iterative goal revisions using a simplified RL agent with simulated top-down grid views. The core idea is for the model to learn to apply a correction to move closer to the intended goal. Since the Blocks World environment is easily simulated, we can automatically generate corrections based on the directional shift needed from a given candidate position to the final goal. We focus solely on the source block prediction sub-task for this case study.

### Experimental Setup

**Environment** We simulate a simplified version of the Blocks World environment on a 10x10 2D board using OpenAI's RL Gym[9] (Brockman et al., 2016). The resulting environment is a top-down projection of the board, similar to the images generated in Section 3.5.2.

**Data** To abstract away from language noise, we automatically generate directional corrections by calculating the shift required to move from the candidate position (or the robot's current position) to the goal. This is inspired by Misra et al. (2017), who use the four cardinal directions output actions, and by Mehta and Goldwasser (2019), who generate simple corrective guidance also in four cardinal directions. Each board region holds three values: a float value representing the region's entropy, which starts high and decreases over the episode, and two binary indicators showing whether the robot or a block occupies the region.

To generate the directional corrections, we first calculate the relative translation (in XY coordinates) between the robot's last action or position and the goal block. Then, we convert this translation into a cardinal direction (e.g., north, southeast). For instance, referencing Figure 3.7 $T1$, if the goal block is at (3,3) and the robot's last action occurred at (1,8), the required translation is east ($3-1=2$) and south ($3-8=-5$), thus generating the correction 'southeast'. Applying this correction reduces the entropy in the regions of the environment from the robot's position (regions with lower entropy are more likely to contain the goal block), effectively guiding the agent and illustrated by the darker pink shade southeast of the blue block in Figure 3.7 $T1$. In the subsequent iteration, the agent uses this information to predict a new position closer to the goal. This cycle repeats until the agent reaches the goal or exceeds the maximum number of iterations. Because this method uses automatically generated directions instead of natural language, it bypasses the difficulty of grounding instructions or corrections to the world, which models in the previous section struggled with.

To simplify the input and abstract away from natural language complexities, we automatically generate directional corrections. First, we calculate the relative translation (in XY coordinates) between the robot's last action or position and the goal block. Then, we convert this translation into a cardinal direction (e.g., north, southeast). For instance, referencing Figure 3.7 $T1$, if the goal block is at (3,3) and the robot's last action occurred at (1,8), the required translation is east ($3-1=2$) and south ($3-8=-5$). We therefore generate the correction 'southeast'. Applying this correction reduces the entropy in the corresponding regions of the environment, effectively guiding the agent (as illustrated by the darker pink shade southeast of the blue block in Figure 3.7

[9]Toolkit for simulating and evaluating RL applications, see `https://www.gymlibrary.dev/`

$T1$). In the subsequent iteration, the agent uses this information to predict a new position. The agent repeats this cycle until it reaches the goal or exceeds the maximum number of iterations. Because this method uses automatically generated directions instead of natural language, it bypasses the difficult challenge of grounding instructions or corrections to the world state—a task the models in the previous section struggled with.

We simulate episodes as needed by randomly sampling a block configuration from one of the Blocks World splits (train/dev/test), setting the true source block as the goal. The episode starts by initializing the environment with this configuration, which is then simulated step-by-step until the episode concludes.

**Task** The objective is to train the model to map user feedback (corrections) for a candidate position to an appropriate action in the environment that follows the correction. For example, if the candidate position is (1,8) and the correction is 'southeast' the predicted action should target a position both below and to the right (east) of the candidate (see Figure 3.7).

A **state**, $s$, is defined by a 3-dimensional array representing the 100 board regions, where each region has an entropy value, $e_{(x,y)}$, and two binary flags indicating the presence of the robot and any blocks. Additionally, each state includes an oracle correction $c$ that is relative to the agent's current candidate position. The corrections $c$ are one of eight possible cardinal and ordinal directions (north, south, northwest, etc.).

At each iteration $i$ the agent predicts an **action** $a_{(x,y)}$ that maximizes a reward function $R(i)$ : $S \times A \rightarrow R$, designed to reduce entropy across the grid as the agent approaches the goal:

$$R^{(i)}(s,a) = \begin{cases} +1.0 & \text{if } a^{pred}_{(x,y)} = a^{goal}_{(x,y)} \\ -1.0 & \text{if } i \geq max\_iterations \\ -\delta/\Delta \text{ Count} & \text{if block at } a_{(x,y)} \\ -\delta & \text{else} \end{cases} \tag{3.2}$$

where $-\delta/\Delta\text{Count}$ represents the difference between the number of blocks with entropy matching the goal in this iteration and the previous one, $i-1$:

$$\Delta\text{Count} = \left|\{(x,y) \in \text{Blocks} \mid e_{(x,y)} = e^{\text{goal}}\}_i\right| - \left|\{(x,y) \in \text{Blocks} \mid e_{(x,y)} = e^{\text{goal}}\}_{i-1}\right|$$

Each episode begins with high entropy across all grid regions. At each iteration, the agent predicts a region to minimise entropy based on a given correction from a rule-based oracle. In this

way, the agent predicts regions that progressively narrow down the goal's location (see Figure 3.7 for a visual example). The reward function encourages actions to align with the correction's direction while maximising the number of blocks 'discarded' through entropy reduction. Actions that reduce entropy across multiple blocks are prioritised over general actions, encouraging the agent to make highly targeted moves.

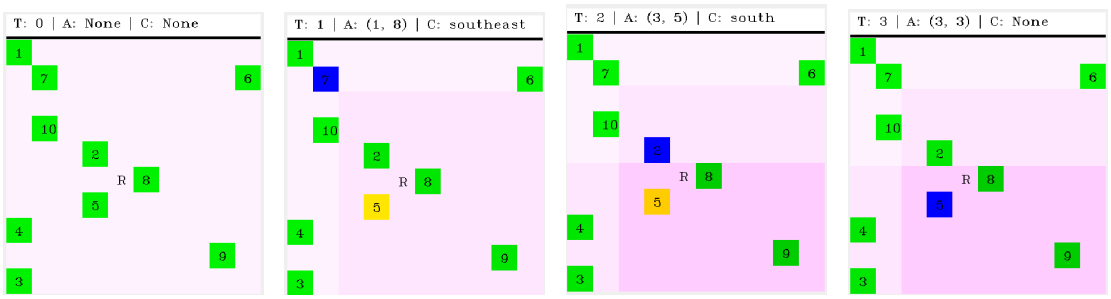



Figure 3.7: Visualisation of the RL agent over 3 iterations, from left (iteration 0) to right (iteration 3). We denote in blue the last action and in gold the true block. At each correction, the entropy in positions of the direction of the oracle correction reduce accordingly. Darker shades of pink indicate lower entropy values on these regions.

**Policy** We model the agent's policy using an actor-critic network from Stable Baselines[10] (Raffin et al., 2019). Both the actor and critic networks share a common feature extractor and each is followed by two fully connected layers with 64 units each.

**Training** We optimise the network with Proximal Policy Optimisation (PPO) (Schulman et al., 2017). PPO uses both an actor network to predict actions and a critic network to estimate state values, providing training signals to directly learn an optimal policy. This is opposed to first estimating state values as in Q-learning (Mnih et al., 2013), which makes PPO much faster to train with fewer samples than many other RL algorithms. We train for 250 epochs and $2 \times 10^4$ iterations each, with a learning rate of $3 \times 10^{-4}$. We evaluate the agent at the end of every epoch with 1000 randomly-generated simulations.

### Results and Discussion

We present the agent's performance in Figure 3.8. The results show that the agent learns a policy to handle corrections effectively, particularly during the initial iterations of each episode. By the second correction, the agent successfully completes ≈40% of episodes, and the mean block distance matches that of the best neural model by Tan and Bansal (2018). As the episode progresses, the agent continues to reduce the mean block distance to the correct source block.

[10]Library of RL models and algorithms, see `https://github.com/DLR-RM/stable-baselines3`

Unlike our previous approaches, this agent demonstrates the ability to follow corrections successfully, grounding them in the environment. However, it is important to note that this environment is simplified and it is not dealing with the variability and nuances of natural language. The state representation also explicitly indicates the position of the robot and blocks, which allows the agent to quickly learn to select blocks without reasoning about the environment or coordinates (e.g., this agent correctly predicts the source block 20% of the time after a single correction, compared to the HCN or CNN models, which achieve $\approx 10\%$).

The improved performance of the RL agent in predicting locations stems from several factors. First, we provide the model with valuable additional information in the world state as entropy, allowing it to distil knowledge from corrections and progressively narrow down candidate locations. Secondly, the RL training allows the agent to learn from numerous interactions, far exceeding those in our supervised learning (Section 3.5), and identify the most rewarding states. Additionally, by modelling the setup as a sequence of steps, the agent develops an understanding of how the world evolves and how to effectively address corrections.

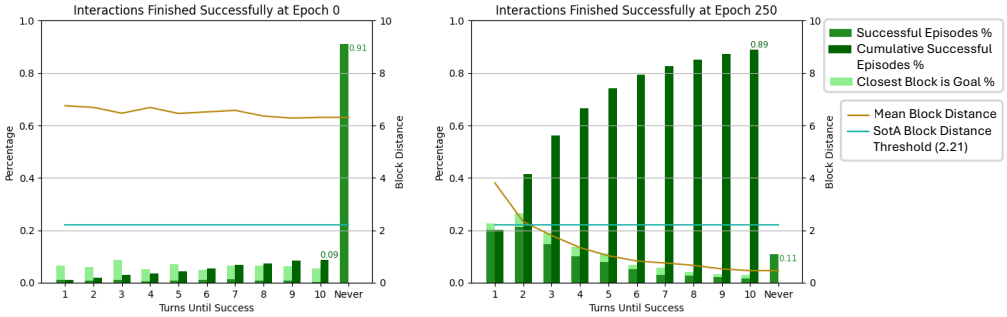
These findings suggest that corrections are indeed valuable in collaborative settings and that it is feasible to train an agent to adjust predictions based on feedback using RL, iteratively narrowing down the final goal. Our agent is not yet able to learn to process feedback, as it is modelled in the world, but is a promising intermediate step towards human-agent collaboration. In Section 6.6, we will explore how to extend this RL approach to VLMs and learn how to follow corrections.

## 3.6 Discussion

This chapter examined situated instruction understanding in collaborative human-robot tasks. Although mutual coordination is crucial in human communication, most prior work on instruction understanding focuses on agents that merely follow instructions without engaging in feedback exchanges to negotiate shared understanding. We therefore investigate ambiguous and underspecified instructions within collaborative dialogues to train models that are robust to miscommunications and failures, making them more suitable for real-world applications.

We begin by introducing the Blocks World (Bisk et al., 2016), a highly ambiguous environment for instruction understanding in which an agent arranges blocks on a table. Previous works have struggled to understand instructions here, which we attribute to the absence of a bi-directional information flow between the human and the agent. Humans would naturally resolve misunderstandings in such ambiguous settings with repairs or corrections. To explore misunderstandings in collaborative dialogues, we collected a corpus of 796 dialogues containing corrections closely


(a) Agent performance over the course of training, comparing mean rewards (↑), mean episode length (↓), successful interactions (↑) and the mean block distance at the end of episodes (↓).




(b) Comparison of agent performance before training (epoch 0, left) and after training (epoch 250, right). We compare the percentage of successful episodes (↑) with the mean block distance (↓) with a maximum of 10 iterations. We use the SotA distance from Tan and Bansal (2018).

Figure 3.8: Evaluation results for the RL agent.

tied to the context and environment based on the Blocks World.

We then explored two model architectures with distinct approaches for instruction understanding within situated contexts. These models, based on HCNs and CNNs, successfully learned to predict actions in these collaborative dialogues, a mix of physical and dialogue intents (e.g., move, clarify) to cooperate in simulated scenarios. However, they significantly struggled predicting positions and adapting to user corrections. We identified that they were unable to ground referring expressions to specific blocks or locations, ultimately limiting their usefulness in collaborative tasks. On the other hand, these models learned to recognise ambiguous referring expressions that needed a subsequent clarification, which we will further investigate in the next chapter (Chapter 4).

Finally, we explored a RL approach to learn to follow corrections through a case study with a simple RL agent. Our findings indicate that corrections could be useful for imperfect agents as, even if their initial predictions are inaccurate, they can further refine them with human input.

However, this setup removed most of the language and vision noise, which are key challenges for training collaborative agents.

The simplified visual environment of the Blocks World presented here allows us to control a wide range of conditions and experiment with underexplored aspects of situated collaborative dialogues. Given its simplicity yet inherent ambiguity, this setting is unlikely to be resolved by larger or more capable models alone, as a deeper understanding of cooperative communication and world states is needed. Our experiments with the corpus produced mixed results: while the models successfully predicted actions (including when a clarification was needed), they struggled with world and spatial reasoning. In this regard, we must also consider that this environment may be challenging even for humans. In Chapter 6, we will explore these dialogues using new approaches for learning in interactive settings and introduce a human study to establish a baseline and analyse model differences. The next chapter focuses on resolving referring expressions, which we identified as one of the primary difficulties for the models in our experiments.

# Chapter 4

# Learning Multi-Modal Representations for Resolving Referring Expressions

This chapter investigates learning to ground objects in the environment, a crucial capability for situated conversational agents. These agents must comprehend their surroundings and accurately identify the objects referenced during a dialogue. To achieve this, we train and evaluate multi-modal vision and language models designed to resolve referring expressions within situated interactions, as well as detect when there is not enough information to uniquely identity an object.

## 4.1 Introduction

Language use in dialogue is highly context-dependent. The meaning of what is said not only depends on the context provided by previous conversational turns; it can also depend on the objects, events, or people in the immediate environment of a conversation. There are many context-dependent phenomena in dialogue, such as fragments, ellipsis and anaphoric expressions. To interpret anaphoric expressions, such as pronouns or definite referential descriptions, their antecedents need to be identified in the context of the dialogue. This context may exist within the dialogue history, or cross-modally in the non-linguistic, visual context of the environment. For example, resolving a pronoun to its antecedent could involve identifying "it" as a pronoun for coat in "*Bring the coat please. It is on the chair*" or linking a phrase like "the blue jacket" to a specific physical or virtual entity in the visual context.

In some cases, referencing the dialogue history and looking at previously mentioned entities may be enough (e.g., "it" for coat). In other situations, it may be necessary to look at the image of the scene and refer to one of the objects there (e.g., "*the blue jacket*").

Often, successful interpretation requires drawing on both the dialogue context and visual attributes simultaneously to discriminate among similar objects.

This chapter explores how models can learn to resolve referring expressions in multi-modal dialogues. Recognising objects and connecting them with their corresponding linguistic expression was one of the main challenges in the previous chapter (Chapter 3), limiting models' ability to collaborate effectively. Here, we design and train a model to address the symbol grounding problem (Harnad, 1990) within situated multi-modal dialogues. We use the SIMMC 2.0 dataset (Kottur et al., 2021), which provides images and dialogues rich in referring expressions within a virtual shop. However, these referring expressions are in highly-ambiguous visual scenes (i.e., with multiple similar-looking objects) and may not be enough to identity the correct objects, leading to miscommunications and subsequent clarifications. Thus, we also experiment with models for detecting ambiguities as a first step towards dealing with miscommunications. Throughout this chapter, we analyse the multi-modal representations models learn for the task of referring expression resolution and ambiguity detection, following the SIMMC 2.0 challenge. Our experiments show the impact of alternative multi-modal representations and the importance of robust models for comprehending the surrounding environment.

## 4.2 Related Work in Referring Expressions

Associating or *grounding* visual objects to their natural language expression has been extensively studied in relation to the symbol grounding problem (Harnad, 1990), often under various terminologies (e.g., referring expression resolution, grounding, understanding; see Kennington et al., 2015; Schlangen et al., 2016; Yu et al., 2016a; Zhang et al., 2018; Cirik et al., 2018), and in this work, we adopt the term REC (Wu et al., 2022). Prior works rely on training models on large referring expression datasets, particularly RefCOCO (Yu et al., 2016a) and its variants, RefCOCO+ (Yu et al., 2016a) and the more challenging RefCOCOg (Mao et al., 2016), which have been crowd-sourced within reference games like ReferItGame (Kazemzadeh et al., 2014) using MSCOCO images (Lin et al., 2014a). REC is also closely related to tasks such as Visual Question Answering (Antol et al., 2015; Goyal et al., 2017), Visual Language-Navigation (Anderson et al., 2018b; Gu et al., 2022) and visual dialogue (Das et al., 2017; de Vries et al., 2017).

Previous approaches for REC typically divide the task into two steps: 1) analysing

images to generate RoIs with object detection models, using CNNs to extract features and classify objects within bounding boxes (Szegedy et al., 2013; Sermanet et al., 2014; Girshick et al., 2014), for example ResNet (He et al., 2016) and Faster-RCNN (Ren et al., 2015); followed by 2) these candidate objects along with their features are combined with the encoded language input to obtain a cross-modal representation, typically through an LSTMs (Nagaraja et al., 2016; Cirik et al., 2018) or, more recently, with attention mechanisms (Anderson et al., 2018a) and transformers (Tan and Bansal, 2019; Chen et al., 2020b; Su et al., 2020).

Most of these approaches operate under the assumption that each input sequence always refers to an object within the image, following their caption-image pre-training. However, they often overlook more general scenarios, such as when a dialogue turn may not reference any objects in the environment (e.g., *"I would like to see blue shirts"* does not explicitly need grounding to the visual scene), or when a reference mentions multiple objects (e.g., *"What is the price of those red and green jumpers?"* refers to two distinct jumpers). Several works address this limitation, such as Lee et al. (2022) who propose an additional binary head to predict zero targets and a secondary loss that heavily penalises object predictions when none are referenced; or Huang et al. (2022), which uses binary heads for each object with a UNITER-based model (Chen et al., 2020b).

## 4.3 The SIMMC 2.0 Dataset

The Situated Interactive MultiModal Conversations 2.0 (SIMMC 2.0) dataset (Kottur et al., 2021) consists of task-oriented dialogues in a multi-modal virtual shop environment. The system agent acts as a shopping assistant to a user responding to questions about items, either clothes or furniture, in a shared shop scenario, e.g., *"What is the price of the red t-shirt?"* or *"Can you add that lamp to my shopping cart, please?"*.

The dataset contains 11,244 English dialogues across two domains (fashion and furniture) collected using a mix of dialogue self-play and crowd-sourcing, and divided into train / dev / devtest / test-std splits with 65% / 5% / 15% / 15% of the dialogues respectively. Table 4.1 summarises the main dataset statistics.

Dialogues take place within virtual scenes populated with various objects, and include annotations for turns (e.g., objects mentioned) and object metadata (e.g., colour or type). Each dialogue is paired with a corresponding scene image, though most dialogues fea-

| | |
|---|---|
| Total # dialogues | 11,244 |
| Total # utterances | 117,236 |
| Total # unique scenes | 1566 |
| Mean # words per user utterance | 12 |
| Mean # words per assistant utterance | 13.7 |
| Mean # utterances per dialogue | 10.4 |
| Mean # objects mentioned per dialogue | 4.7 |
| Mean # objects in scene per dialogue | 19.7 |

Table 4.1: SIMMC 2.0 dataset statistics, from Kottur et al. (2021).

ture multiple images from different viewpoints as they progress, introducing new objects into focus. The combination of long, context-dependent dialogues and varying perspectives requires models to retain and carry over information from earlier turns and across modalities. The example dialogue 10 below provides a change of perspective during the conversation.

*(10)* Excerpt of a SIMMC 2.0 dialogue with a perspective change within the virtual scene. After several turns, the perspective (image) changes, revealing new objects and hiding others. Some hidden objects are still referenced, requiring models to remember previous objects.

**User:** *I'd like the size and rating for the pants.*

**System:** *Which pair are you asking about?*

**User:** *I mean the gray ones in the second cabinet.*

**System:** *Those are an XS, and have a 3.5 rating.*

**User:** *Sounds great, I'll go with those.*

**System:** *Nice choice, I'll add them to your cart.*

...

**User:** *I'm also interested in taking a look at some half-sleeve shirts.*

**System:** *In the bottom middle of the left wall I have that black shirt, as well as the pink and white striped shirt in the middle cabinet on the right wall.*

**User:** *Have you got anything like the black and olive dress at the far left of the dress stand, but made by New Fashion?*

...

The dataset released as part of a challenge[1] with four sub-tasks: ambiguity detection, coreference resolution, dialogue state tracking and response generation and retrieval. The test-std split is used for challenge benchmarking and its true prediction targets are not public, hence this chapter's experiments use the public devtest split for evaluation. The following sections present research done as part of our submission to the first two sub-tasks, disambiguation detection and coreference resolution.

## 4.4 Learning to Resolve Referring Expressions

Understanding shared references in dialogues is essential, yet highly-context dependent. This often involves resolving anaphoric expressions, such as linking pronouns to their antecedents (e.g., *"Bring the coat please, it is on the chair"* or identifying physical/virtual entities around us (e.g., *"the black and olive dress at the far left"*). The dialogue context may sometimes be sufficient to understand mentioned entities. In other cases, it may be necessary to examine the surroundings to locate the referred object (e.g., *"the black and olive dress"* in example 10). However, more often than not, a combination of both dialogue and visual contexts is required to accurately identify an entity.

This sub-task of the SIMMC 2.0 challenge, **Multi-Modal Coreference Resolution**, requires models to resolve referring expressions in user dialogue turns to their corresponding objects in the scene. This process encompasses both grounding descriptive visual references (e.g., *"What is the price of the red jacket?"*) and resolving textual pronominal references (e.g., anaphoric references, *"What is the price of it?"*). The goal is to determine the specific object or set of objects an expression refers to within a user-agent dialogue. While traditional coreference resolution solely focuses on establishing links *within* the text (e.g., linking "it" back to "the red jacket")[2], this task combines this objective with visual grounding in a multi-modal context[3]. When a model correctly grounds multiple expressions (like "the red jacket" and later "it") to the same visual object, it resolves their coreference via this shared visual link. Therefore, the mechanism heavily relies on grounding language to perception (or symbol grounding (Harnad, 1990),

[1] See The Tenth Dialog System Technology Challenge (DSTC10) [last accessed 2024-10-12].

[2] This task shares similarities with Entity Linking (Ji and Grishman, 2011), which SIMMC 2.0 also incorporates by requiring models to link entities to their non-visual metadata (e.g., *"Can you show me jackets from the YogiFit brand?"*).

[3] Also see Referring Expression Comprehension (Yu et al., 2016a; Hu et al., 2015)

discussed in Section 2.1.2), but the task fundamentally aligns with coreference resolution because it requires identifying the scene-specific referent for each expression. Henceforth, we adopt the term coreference resolution, as following the name of the SIMMC 2.0 sub-task, while acknowledging that it covers both resolving textual references and performing visual grounding or REC. Models must leverage the current utterance, prior dialogue and visual information to perform this grounding, predicting the specific object(s) mentioned in the user's utterance, or an empty set if the utterance contains no references.

### 4.4.1 Task Formulation

We formally define the task as a multi-label classification problem. Given a user utterance at turn $t$, $u_t$, the prior dialogue history as a sequence of alternating system and user utterances $H = [(u_1, s_1), ..., (u_{t-1}, s_{t-1})]$, an image $I_t$ and a set of objects in the scene $O_s = \{o_1, ..., o_k\}$, the task is to predict the subset of objects $O_{ref} \subseteq O_s$ referenced in $u_t$. Models have to approximate Equation (4.1).

$$f(u_t, H, I, O_s) = O_{ref} \tag{4.1}$$

The scene provides details about visible objects, including their bounding boxes in the images and their IDs (prediction targets). These object IDs can be linked to metadata to extract both visual properties (e.g., color) and non-visual attributes (e.g., price). While visual metadata is not available during evaluation, it can be leveraged during model training.

### 4.4.2 Approach

As the SIMMC 2.0 dialogues require deep integration of visual and textual contexts within a multi-label classification task (user turns may reference 0, 1 or more objects), we introduce a new model architecture to learn visual object attributes and exploit them in both the visual and textual contexts.

In our model pipeline, we begin by extracting visual features using an object detection model. We then encode these features, along with the language inputs, with a multimodal model, followed by a custom decoder component. We discuss each stage in detail

separately below, with the full architecture outlined in Figure 4.2.

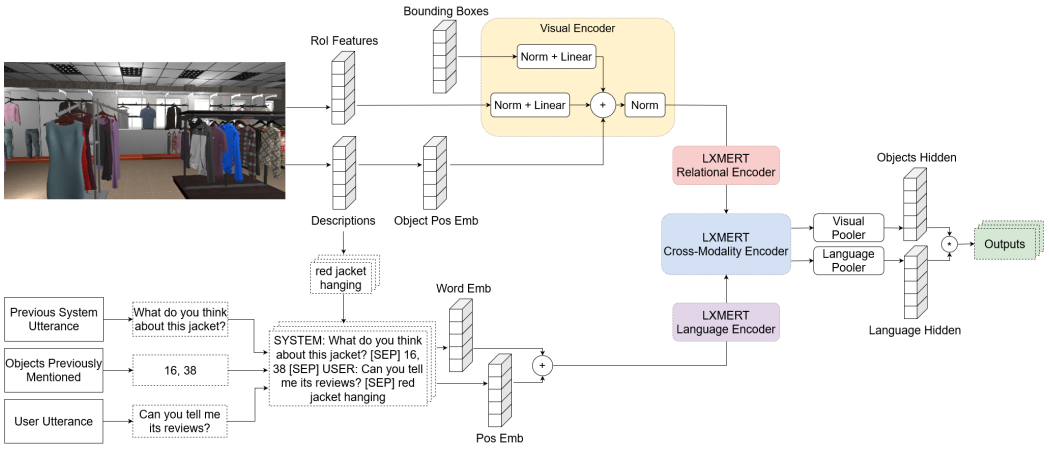



Figure 4.2: Model architecture for multi-modal coreference resolution. It uses vision to derive textual descriptions, and it combines the RoI features with language to obtain the final outputs. 'Norm' and 'Linear' stand for normalisation and linear layers.

### Visual Feature Extraction

We use Detectron2[4] (Wu et al., 2019) to train an object detection model for the visual feature extraction. Specifically, we fine-tune a Faster-RCNN (Ren et al., 2015) with a ResNet+FPN backbone. This model is originally pre-trained on MSCOCO (Lin et al., 2014a) for instance segmentation, which enables to detect each object individually within a section of the image. This is particularly useful in SIMMC 2.0 scenes, where the same object class (e.g., t-shirt) may appear many times in different locations or with different attributes. On the other hand, MSCOCO images are photo-realistic so the model struggles to perform well in the simulated environments of SIMMC 2.0 without further adaptations. Below, we outline the data preparation and fine-tuning steps required to optimise this model for our task.

**(1) Data Preparation** Using the object metadata from SIMMC 2.0, we derive object labels by combining each object's colour, asset type (e.g., jacket, t-shirt) and state (e.g., folded), resulting in *object description strings* (e.g., 'red jacket hanging'). For objects in furniture scenes, which do not have states, we simply use their type (e.g., 'brown sofa').

[4]An open-source library for object segmentation and detection models, see `https://github.com/facebookresearch/detectron2`

We also simplify composite colours, such as converting 'light blue' to 'blue'.

In total, we obtain 17 colours and 21 object types, creating 169 unique combinations that appear in the dataset. These combinations represent the full set of object labels, which serve as the prediction targets. The final dataset for training the vision component consists of SIMMC 2.0 images, the bounding boxes for all objects within them, and the derived object labels.

**(2) Fine-tuning** We follow the training regime outlined in Wu et al. (2019) and fine-tune the model with a multi-task objective, jointly optimising both bounding box object detection and object classification. Object bounding boxes are available during inference as part of the scene information in the coreference resolution task, however, we still train the model to detect them, as this significantly improves overall performance (boosting mean accuracy on the devtest set from 41.3% to 65.4%). Training for bounding box prediction helps the model learn where to focus and what salient object features to prioritise. For object classification, we use the object description labels (e.g., "red jacket hanging") as the prediction targets. We fine-tune the model using the images from the SIMMC 2.0 train set, and evaluate its performance on the dev set to select the best-performing model.

**(3) Extraction** Since the object detection may miss some objects in the image, we provide the model with all the gold bounding boxes after the detection step to force a label prediction for each object. We also modify the model to expose its final layers, allowing us to extract the Region of Interest (RoI) features to use in our vision encoder. Ultimately, for each object in the image, we obtain both a RoI feature vector and a description label. We run the extraction in all dataset splits with the final fine-tuned model, so we can use visual features to train our multi-modal encoder in the next step.

### Multi-Modal Encoder

We use the LXMERT model (Tan and Bansal, 2019), pre-trained on multiple vision and language datasets, as our multi-modal encoder. Built based on the transformers architecture (Vaswani et al., 2017), LXMERT employs a dual stream encoder-decoder approach that first processes vision and language separately through dedicated modules, then combines them using cross-modal attention layers to generate multi-modal contextual representations. The model outputs separate representations for both modalities (vision and

language), along with a cross-modal hidden state using the special [CLS] token, as described in (Devlin et al., 2019). Figure 4.3 shows LXMERT's architecture.

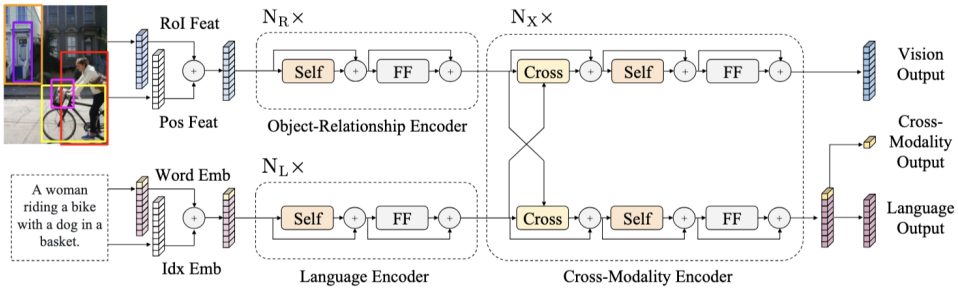



Figure 4.3: LXMERT (Tan and Bansal, 2019) architecture.

LXMERT outputs a single cross-modal state, which is well-suited for tasks like visual grounding, VQA or Natural Language Inference (NLI), where classifying a single object or answering yes/no questions is sufficient. However, this makes it less suitable for our coreference resolution task, which requires predicting a list of 0, 1, or more objects. Additionally, LXMERT is not pre-trained on dialogue data, limiting its ability to handle long-context conversations effectively. We overcome these limitations with a custom architecture in the next section.

### Custom Model Architecture

To handle more than one object at a time in dialogues, we use the LXMERT encoders and extend it to handle multiple input sequences. We additionally add cross-sequence poolers to obtain multiple outputs, one for each object, as shown in Figure 4.2. We describe this in detail below.

**Input** The LXMERT encoder requires two inputs: a sentence in natural language and visual features:

(i) **Sentence:** For each user dialogue turn, $u$, we construct a language sequence $l$ by concatenating the dialogue history $H$ and the list of object IDs previously mentioned by the system in each turn $O_{t-1}$: $concatenate(u, H, O_{t-1})$. To distinguish between system and user turns, we insert the special token [SEP] between turns, and prepend the prefixes SYSTEM and USER following practices from dialogue-oriented models (e.g., (Wu et al., 2020a)). We also limit the dialogue history to a maximum number of 2 prior turns. The final language sequence is shown in Equation (4.2)

and example 11. Finally, we append the object description labels extracted in Section 4.4.2 for each object $O_s$ in the scene, $d_k$, obtaining one sequence per object, $L = \{l_1, ..., l_k\}$. We denote everything in the sequences up to the last `[SEP]` as the context, and $d_k$ as the question with segment IDs, similar to the tasks of sentence entailment or QA (Devlin et al., 2019).

$$l_k = \text{[CLS]}...\text{[SEP]}s_{t-1}\text{[SEP]}O_{t-1}\text{[SEP]}u\text{[SEP]}d_k \tag{4.2}$$

*(11) SYSTEM : We have this red, white and yellow blouse on the left, and the white blouse on the right [SEP] 60 56 [SEP] USER : I'd like to get that blouse, please [SEP] 25 white blouse hanging*

(ii) **Visual features:** we use each object's RoI features, $F_t = \{f_1, ..., f_k\}$, along with their bounding box coordinates, $P_t = \{p_1, ..., p_k\}$, to learn position-aware embeddings and spatial relationships in the images. To handle the presence of multiple objects of the same or similar type in SIMMC 2.0 images, we introduce positional object IDs that represent the sequence of objects in the image–similar to word position IDs (e.g., the first jacket is assigned 1, the second one 2, etc.), $E_t = \{e_1, ..., e_k\}$. These positional IDs enable the model to better distinguish between objects of the same type (and potentially the same colour).

**Encoding** The model first tokenises the text sequences, $L$, and extracts BERT embeddings. The visual encoder then combines the visual features into a visual feature state as follows:

$$\begin{aligned}
\hat{f}_k &= LayerNorm(W_f f_k + b_f) \\
\hat{p}_k &= LayerNorm(W_p p_k + b_p) \\
\hat{e}_k &= PositionEmb(W_e e_k + b_e) \\
\hat{v}_k &= Dropout(LayerNorm(p_k + f_k + e_k)))
\end{aligned} \tag{4.3}$$

The sentence embeddings and visual features are processed through the standard LXMERT encoders, which include language, relational and cross-modality layers, to

obtain two feature sequences with shared modality information: one language $\hat{h}_k^l$ and one vision $\hat{h}_k^v$. We then apply pooling and compute the dot product to extract a hidden cross-modal hidden representation $\hat{h}_k^x$. This is then passed through a sequence of GeLU, normalisation and fully connected linear layers to produce the final output representation, $\tilde{h}_k^x$.

$$\begin{aligned} \hat{h}_k^l, \hat{h}_k^v &= LXMERT^{encoder}(\hat{l}_k, \hat{v}_k) \\ \hat{h}_k^x &= LayerPool(\hat{h}_k^l) \cdot LayerPool(\hat{h}_k^v) \\ \tilde{h}_k^x &= (W \cdot LayerNorm(GeLU(\hat{h}_k^x)) + b) \end{aligned} \tag{4.4}$$

**Output** Since the prediction target is a list of object indices corresponding to those referenced in the user turn $u$, the model outputs a vector of values between 0 and 1, where each element represents the probability that a specific object was referenced in $u$. We formulate the task as a multi-label classification problem:

$$Probability(k) = Sigmoid(\tilde{h}_k^x) \tag{4.5}$$

The predicted object IDs are selected when their probability score is above a threshold, so the model may predict multiple or no objects. We used the validation set to define the best threshold value, usually this is a value between 0.3 and 0.4.

**Training** Given that the model inputs consist of $k$ sequences, one for each object, we quickly encounter computational limitations due to both the number of model parameters (layers) and the objects per scene. To mitigate this, we implement the following strategies:

- **Layer reduction:** we first reduce the number of layers in the LXMERT encoders from the original configuration (9 in language, 5 relational and 5 cross-modal) to a more compact 5-3-3 layers, respectively. Our initial evaluations indicated that combinations with fewer layers yielded significantly lower performance, while those with additional layers (e.g., 7-4-4) did not offer substantial improvements relative to the increased computational cost.

- **Token sequence truncation:** we limit the size of the input text sequences to 200 tokens each per object, which is usually enough for 2 dialogue turns. We truncate

sequences exceeding this limit by removing the oldest dialogue turns first.

- **Training on small scenes:** The SIMMC 2.0 scenes average 27.6 (SD 20.7) objects, reaching up to 141 objects. Since for each input object we also pass to the model its visual RoI features and a text sequence, $l_k$, we quickly reach prohibitive memory demands during batched training in scenes with many objects Consequently, we reduce the size requirements to train the model by excluding all dialogues associated with scenes containing more than 60 objects from training, which accounts for approximately 4% of the train data. This measure limits the maximum object count $k$ during training to 60 objects, which allows us to pad batches to this fixed size. Since the model processes objects individually until the final pooling layers, this training approach allows for effective learning on smaller scenes while enabling scaling to the full range of object counts during evaluation without significantly compromising performance.

We train for a total of 10 epochs minimising the Binary Cross-Entropy (BCE) loss for each object $k$ separately.

**Limitations** Our model architecture enables individual attention and reasoning for each object; however, this imposes constraints on the maximum token sequence size and the number of layers we can use due to hardware limitations. Additionally, the model is capable of handling coreferences involving multiple objects, but it is unable to refer to the same object multiple times. This model architecture and task design in particular cannot handle these cases, which are a very rare possibility in SIMMC 2.0 dialogues. For instance, in the utterance *"USER : I'll take two of the wood one"*, the user refers to two identical objects, resulting in the prediction target `[7, 7]`.

### *4.4.3 Experimental Setup*

In this section, we evaluate our model in the task of coreference resolution with the SIMMC 2.0 dialogues.

#### Metrics

In line with the SIMMC 2.0 evaluation protocol (Kottur et al., 2021), we assess coreference resolution performance using **Object F1** (also referred to as $F$-score). This metric

represents the harmonic mean of recall and precision for the predicted objects at each dialogue turn, see Equation (4.6). Since SIMMC 2.0 dialogue turns may reference 0, 1 or more objects, the Object F1 score effectively summarises model performance as a multi-label classification task, with higher scores indicating better performance.

$$Object\ F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.6}$$

**Baseline**

We compare against the language-only baseline provided in Kottur et al. (2021). The model is based on GPT-2 (Radford et al., 2019) fine-tuned on all the SIMMC 2.0 challenge tasks that generates object IDs using special multi-modal tokens `<SOM>` and `<EOM>` from the user utterance and dialogue history. By default, the model uses the previously mentioned object IDs in the input with the special tokens. We report the F1 score for our reproduction of the baseline, 0.2% lower than reported in their paper.

### *4.4.4 Results*

Table 4.2 presents the performance of our model for coreference resolution (row 1), along with ablations of various components. All model ablations were trained with the same hyperparameters for 10 epochs and evaluated on the devtest split. Overall, our findings indicate that language is the most critical feature for our model. We discuss the ablation results in detail below.

**Vision** Ablating vision-related components reveals small negative impacts, indicating that they have a limited effect on the model's performance. RoI features (row 5) emerge as the most significant, with their removal causing a 3.39% decrease in object F1 compared to the full model. Interestingly, ablating bounding boxes (row 6) does not affect the model performance despite their importance in spatial reasoning and their role in LXMERT pre-training (Tan and Bansal, 2019). This suggest that bounding boxes may be too noisy in most scenes or lack the fine-grained details necessary to discriminate objects, which is a crucial requirement in this dataset with multiple overlapping objects. Object position IDs (row 7) provide a slight benefit, likely aiding in scenes with many objects. However, they rely on the object descriptions from our visual extraction (Section 4.4.2) and hence may

| | Coreference Model Performance | Object F1 ↑ | Δ |
|---|---|---|---|
| 1. | Our Full Model | 60.83 | - |
| 2. | Baseline (Kottur et al., 2021) | 36.37 | -24.46 |
| 3. | no Object IDs | 15.23 | -45.60 |
| | **Vision** | | |
| 5. | no RoI features | 57.44 | -3.39 |
| 6. | no bounding boxes | 60.83 | 0 |
| 7. | no object position IDs | 60.78 | -0.05 |
| 8. | no visual or cross-modal encoders | 57.64 | -3.19 |
| | **Language** | | |
| 9. | no user utterances | 33.72 | -27.11 |
| 10. | no previous system turn | 60.74 | -0.09 |
| 11. | no previous objects | 48.93 | -11.9 |
| 12. | no descriptions | 54.50 | -6.33 |
| 13. | no colours in descriptions | 57.36 | -3.47 |
| 14. | no types in descriptions | 57.52 | -3.31 |
| 15. | no object IDs in descriptions | 48.98 | -11.85 |
| 16. | with oracle descriptions (upper) | **70.71** | +9.88 |

Table 4.2: Model performance for coreference resolution. We compare with the baseline from Kottur et al. (2021), with and without using special token IDs. The difference, Delta Δ, is in respect to the full model (Row 1). We mask with 0s the ablated input features.

be inaccurate or too noisy.

In row 8, we remove all connections to the visual, relational and cross-modal encoders of LXMERT, relying solely on the language encoder. We also replace the language pooling layer with a fully connected layer to maintain comparability to other ablations. The performance drops 3.19%, lower than just removing the RoI features, suggesting that the model can develop stronger language representations when relational and cross-modal information is not available.

To better understand the limited effectiveness of RoI features in our task, we further analyse their characteristics. We compute the similarity scores across objects with similar descriptions, obtaining a mean similarity of 0.62 (SD 0.14). This high similarity suggests that RoI features are generally close to each other and therefore may not provide sufficient distinctiveness for the model to accurately identify specific objects. We also use t-SNE (Van der Maaten and Hinton, 2008) to project the representations in a shared embedding space. The resulting visualisation shows that RoI features mainly cluster by class type, which can hinder performance when it comes to discriminating the target object in cluttered scenes. The RoI features seem to capture broad categorical information (e.g.,

jacket, jeans), but are limited when identifying distinct objects of the same class.

This high similarity indicates that the features capture the general appearance of objects, but not their location, primarily for two reasons. Firstly, the feature extraction process, involving a Detectron2 model pre-trained on diverse object classes (Wu et al., 2019) (e.g., horse, cat, person, vehicle) and fine-tuned on SIMMC 2.0 object types and colours, inherently biases the RoI features towards capturing strong categorical signals (e.g., identifying an item as 'jeans' or 'jacket'). Consequently, visually similar items belonging to the same class tend to have highly similar feature representations. Secondly, the locational information encoded within these features is often less distinct, partly because the SIMMC 2.0 dataset frequently features scenes with dense clusters of similar objects (like multiple shirts on a clothing rack), leading to RoI features with overlapping or 'noisy' spatial information.

**Language** Language features are the most critical for the model, particularly the current user utterances. Unsurprisingly, these utterances serve as the most significant important component, demonstrating that our model heavily relies on the user input to identify the referenced objects. On the other hand, previous system turns offer limited utility, and only decrease object F1 slightly when not present. The list of previously mentioned objects, a sequence of object IDs that the system mentioned in the last turn, play a key role, as its ablation leads to a notable drop in performance. This is expected given that previously mentioned objects are more likely to be referenced again within the same dialogue.

Regarding object descriptions (Table 4.2, lines 12-15), we observe large effects when ablating these features. These descriptions are natural language sequences derived from our visual extraction process (Section 4.4.2), which share the same model as the RoI features. Thus, in some way, these serve as a human-readable encoding of the visual RoI features. Unlike the vision ablations, the model learns to use these descriptions effectively for resolving coreferences, despite their extraction inaccuracies (only predicting 65.4% of object descriptions accurately). They enable the model to discriminate objects with distinct properties (e.g., "*blue jacket hanging*" vs "*red jacket hanging*"), which is particularly beneficial in scenes containing many similar objects where only a few properties distinguish each other. Both colour and object type seem to be equally important (Table 4.2, rows 13 and 14), as removal of either leads to a performance drop roughly equivalent to half of entirely removing descriptions. Providing oracle (perfect) object

descriptions (see Table 4.2, row 16) boosts performance by 9.88%, indicating that the model has yet to reach its full potential and that enhancements in the visual extraction could significantly improve conference resolution capabilities.

Descriptions also include the object ID, which is likely essential for the model to learn better representations. These ID complement the object properties and serve as a strong signal to the model. By relating to previously mentioned object IDs, the model can exchange information with the context, thereby reinforcing the signals needed to accurately predict the referenced objects. The GPT-2 baseline (Kottur et al., 2021) heavily relies on object IDs, as seen when ablating them (row 3). Removing previously mentioned object IDs from the input (e.g., `<SOM>56 32<EOM>`), significantly reduces performance from 36.37% to 15.23% object F1. There are two items for discussion with regards this baseline. First, neither version of the baseline model is aware of the total number of objects or their IDs in the scene. Unlike our model, aware of the total number of objects $k$, the baseline only uses dialogue history, user utterances and system-referenced objects to predict object IDs. Since the IDs are pseudo-randomised between scenes and do not start sequentially, an object F1 of 15.23% is notable given the possibility of generating invalid IDs. Second, the baseline appears to learn how to relate these object IDs to dialogue and user utterances, despite the fact that IDs are not unique (e.g., ID 0 could be a red jacket in one scene and a wooden table in another). Interestingly, only 51.6% of the predicted object IDs match previously mentioned objects, with the remaining 48.4% likely hallucinated or linked to some underlying object representation the model learns, akin to the ablated baseline's 15.23%. We will discuss the impact of using object IDs to learn representations in the next section.

### *4.4.5 Task Analysis and Discussion*

This section further analyses coreference resolution in SIMMC 2.0 dialogues to understand how language-only models achieve high scores while also examining the formation of multi-modal representations. We explore the emergent emergent properties that arise due to the SIMMC 2.0 data.

**Additional Baselines** For our discussion, we provide the following additional baselines beyond the one provided by Kottur et al. (2021) (see Table 4.3):

- **Baseline random:** select a single object ID randomly from the scene objects.
- **Baseline random (IDs below 5):** select a single object ID randomly from the scene objects, limiting the the value to 5.
- **Baseline previous objects:** use the system-referenced objects of the last turn as the predictions. We use as many IDs as the system mentioned, as we evaluate with object F1 score.
- **Baseline all previous objects:** use the system-referenced objects as the predictions from all previous dialogue turns.

We also experiment with modifications to our model, discussed in detail below.

| | Approach | Object F1 ↑ |
|---|---|---|
| | **Additional Baselines** | |
| 1. | Baseline random | 0.65 |
| 2. | Baseline random (IDs below 5) | 4.69 |
| 3. | Baseline previous objects | 33.16 |
| 4. | Baseline all previous objects | 32.15 |
| | **Our Model** | |
| 5. | Default | 60.83 |
| 6. | with all previous objects | 63.76 |
| 7. | duplicated objects removed | 71.69 |
| | **Global Object Representation** | |
| 8. | Default + Global IDs | 68.53 |
| 9. | with all previous objects | 68.94 |
| 10. | no descriptions | 70.29 |
| 11. | no RoI features | 66.72 |
| 12. | oracle descriptions (upper bound) | **70.53** |

Table 4.3: Coreference resolution experiments with other baselines and modifications to our model. In row 7, we remove objects with the same properties from the scene, indicating that identical objects in SIMMC 2.0 scenes are a challenge for models.

**Object IDs** The previous Section 4.4.4 suggested that object IDs provide strong signals for models to learn to solve the task. These IDs are not unique across SIMMC 2.0 scenes and were assigned at random, often starting from a non-zero values. A data analysis reveals that they mirror the overall object distribution in the data—for instance, object ID 0 is a jacket in 36% of scenes, a blouse in 18%, and so on. Despite their non-sequential nature, object IDs are also heavily skewed towards smaller numbers, particularly for prediction targets. Specifically, 52.5% of coreference prediction IDs are below 10, and 30.1%

are below 5. As a result, during fine-tuning, models tend to prioritise smaller IDs and may overlook densely populated scenes. This is evident in Table 4.3, where the random baseline restricted to IDs below 5 (Table 4.3, row 2) significantly outperforms the standard random baseline.

**Dialogue History and Previously Mentioned Objects** We observe strong performance by using only the system-mentioned objects in the previous turn $t-1$ (Table 4.3, row 3), suggesting that much of the context to resolve coreferences lies within the dialogue history. However, the system utterances themselves seem less important, as our model struggles to exploit them effectively (as shown in Table 4.2). Providing the model with all previously mentioned objects (from all system turns, not just $t-1$) further boosts its performance, in contrast to the baseline using the same objects (row 4). This indicates that our model can leverage object IDs to learn robust, context-aware representations across dialogue turns. Overall, this suggests that this coreference resolution task (or REC) relies on a combination of previously mentioned objects, the current user utterance and some visual information, but not necessarily the system's past utterances.

**Exploiting Object Information** We experiment with other ways of exploiting object information by creating object representations at the dataset level. SIMMC 2.0 has a limited set of unique object assets (with the same properties) replicated across scenes in different positions. Thus, we leverage this by deriving global tokens to represent each object using metadata. For example, a "red jacket hanging" is mapped to `[OBJ_321]`, which we substitute whenever this particular object is referenced. This approach effectively amplifies the signal models exploit through object IDs and enables the creation of more robust latent object representations. As shown in Table 4.3 (row 8 onwards), global representations improve model performance. We also see an improvement when removing object descriptions (Table 4.3, row 10), but gains with oracle descriptions (Table 4.3, row 12). This indicates that inaccurate descriptions can be detrimental, while accurate ones convey crucial information. Notably, visual RoI features still provide key details missing from these tokens, such as spatial relationships. In a way, these tokens help the model implicitly learn object properties (e.g., recognising that an object with ID X is consistently described as red), similar to disentangled object representations (Locatello et al., 2019). Overall, these findings suggest that the special tokens strengthen the model's abil-

ity to learn more robust object representations, improving its differentiating ability and thus its coreference resolution skills.

## 4.5 Detecting Ambiguous Coreferences

In dialogue, speakers produce turns spontaneously and in real-time, often within dynamic environments, and where there is potential mismatch between the perspectives of the speaker and hearer (see e.g., Dobnik et al. (2015)). As a result, referring expressions may not always uniquely identify the intended referent, leading to referential ambiguities. These ambiguities need to be resolved through subsequent clarificational exchanges (Purver, 2004) for repairing the miscommunication (Purver et al., 2018). However, to do this, hearers need to first identify ambiguities.

We use SIMMC 2.0's **Multi-Modal Disambiguation** sub-task to explore the ability of models to detect ambiguities in the dialogue. Ambiguities can arise from various sources or reasons, such as underspecified referring expressions that fail to single out a unique referent (e.g., *"What is the price of that t-shirt?"*. As with the previous task in this chapter (Section 4.4), models can leverage the array of information and annotations available in the SIMMC 2.0 dialogues. The goal is to determine whether a user's referring expression is ambiguous, so that the system can ask a clarification question in the follow-up turn (e.g., *"Which t-shirt do you mean?"*). Identifying these ambiguities is a crucial initial step in addressing miscommunication in collaborative situated dialogue.

This section outlines our approach to the task and discusses key findings. However, unlike our custom model developed for coreference resolution, we observe that detecting ambiguities in SIMMC 2.0 is comparatively straightforward for models. Thus, we experiment with several model architecture and focus on analysing why models excel in this task whilst struggling with coreference resolution.

### *4.5.1 Task Formulation*

The SIMMC 2.0 dataset generated dialogues through a combination of self-play and crowd-sourcing. First, a simulator produced the dialogue structure by sampling dialogue acts from a probability distribution and using previously generated scenes and object metadata. This process is similar to agenda-based dialogue simulation (Schatzmann et al.,

2007). During this process, ambiguous referring expressions were deliberately introduced into the dialogue flow according to the source of the ambiguity: visual (e.g., *"What's the price on the blue one?"* in scenarios with multiple blue objects) or dialogue-based (e.g., *"What's the price?"* in contexts where multiple objects had been previously mentioned). Turns containing these ambiguities were labelled accordingly and were followed by a clarification question in the subsequent turn. Afterwards, crowd-workers paraphrased the dialogue acts and associated information to produce the final dataset dialogues.

We formally define detecting ambiguous references in SIMMC 2.0 as a binary classification task. Given a user utterance at turn $t$, $u_t$, the prior dialogue history as a sequence of alternating system and user utterances $H = [(u_1, s_1), ..., (u_{t-1}, s_{t-1})]$, an image $I_t$ and a set of objects in the scene $O_s = \{o_1, ..., o_k\}$, the task is to predict if the referring expression in $u_t$ is ambiguous, $\hat{y}_t \subseteq \{0, 1\}$. Models have to approximate Equation (4.7).

$$f(u_t, H, I, O_s) = \hat{y}_t \tag{4.7}$$

### *4.5.2 Experimental Setup*

This section presents the models used for fine-tuning and evaluation, followed by the metrics to used to assess their performance.

#### Models

Our focus is on models that integrate either dialogue or visual contexts to detect multimodal ambiguities effectively.

**Language-Only Models** We fine-tune multiple versions of TOD-BERT (Wu et al., 2020a), a model pre-trained on task-oriented dialogues. It processes dialogue history by concatenating system and user turns sequentially, using special tokens (`[SYS]` and `[USR]`), forming a flat sequence. The final dialogue representation is derived using the `[CLS]` token. We also fine-tune a DialogGPT (Zhang et al., 2020) and RoBERTa (Liu et al., 2019) models with similar configurations. We modify these models to output a binary value applying a Sigmoid function to the final `[CLS]` token. We fine-tune them using BCE loss for 10 epochs, without providing any objects or visual context (e.g., images or RoI features).

**Vision-and-Language Models** To explore the role of vision, we experiment with a base LXMERT model (Tan and Bansal, 2019). As the model is not pre-trained on dialogues, we only provide the user utterance and omit any dialogue history. We extract visual features with a trained Detectron2 (Wu et al., 2019) model (see Section 4.4.2) and provide a list of objects in the scene. We fine-tune for 10 epochs and use its default classification head to predict the binary label.

**GPT-2 Baseline** We also fine-tune and evaluate the language-only baseline from Kottur et al. (2021). It is based on a GPT-2 model (Radford et al., 2019) fine-tuned on detecting ambiguous user referring expressions using the last 5 dialogue turns. Through hyper-parameter tuning, we achieve significant performance improvements over the originally reported accuracy of 73.9%, which we also report.

**Metrics**

Following the SIMMC 2.0 evaluation protocol (Kottur et al., 2021), we assess disambiguation performance using **accuracy**, see Equation (4.8) below.

$$\text{Accuracy} = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives} \quad (4.8)$$

### *4.5.3 Results*

As shown in Table 4.4, models effectively detect ambiguous utterances regardless of the input. Language-only models excel, even without leveraging dialogue history– TOD-BERT-jnt, the top-performing model, achieves this without using prior turns. While LXMERT also performs well, incorporating vision does not seem to offer an advantage, even though ambiguities often arise due to similar-looking objects. In this task, the visual information in SIMMC 2.0 dialogues may be too noisy for learning representations required to detect multi-modal ambiguities, or at least noisier than text. The rest of this section explores these results further and analyses the ambiguities in SIMMC 2.0.

| Model | Previous Turns | Accuracy ↑ |
|---|---|---|
| Baseline Random | - | 50.1 |
| **Language-only** | | |
| GPT-2 Baseline (Kottur et al., 2021) | 5 | 71.0 |
| GPT-2 with optimal hyperparameters | 5 | 88.0 |
| DialoGPT$_m$ (Zhang et al., 2020) | 5 | 91.3 |
| ROBERTA$_{base}$ (Liu et al., 2019) | 5 | 90.3 |
| TOD-BERT BERT-base (Wu et al., 2020a) | all | 91.5 |
| TOD-BERT BERT-large (Wu et al., 2020a) | 5 | 89.6 |
| TOD-BERT-jnt (Wu et al., 2020a) | 0 | **91.9** |
| **Vision+Language** | | |
| LXMERT (Tan and Bansal, 2019) | 0 | 89.4 |

Table 4.4: Ambiguity detection experiments with several models. We show the best version of each model from the dev split and evaluated on devtest. When 'Previous Turns' is 0, we only use the current user utterance as input.

### *4.5.4 Task Analysis and Discussion*

We analyse several key factors related to the task to understand why models perform so well.

**Data and Dialogue History** The SIMMC 2.0 dataset structures the Multi-modal Disambiguation Task such that each dialogue typically contains two user turns with explicit disambiguation labels. Models have to predict whether the turn has a `True` (ambiguous) label and thus the system needs to disambiguate with a clarification request in the follow-up turn, or a `False` (unambiguous) label so dialogue can continue. Our analysis of the label distribution indicates that they are evenly distributed across the dataset splits, with approximately 50% true and 50% false, preventing models from exploiting majority biases.

A disambiguation (or repair) arises after a mismatch in the participants' expectations during a conversation. Thus, it is reasonable to expect that dialogue history could signal such situations, along with other phenomena like the number of discourse entities–where more entities might indicate a higher likelihood of disambiguation. However, including the preceding turns to those that might trigger or avoid disambiguation does not seem to significantly boost model performance. As shown in Table 4.4, even model architectures not specifically trained on dialogue data perform competitively with those that use more context. Our experiments above suggest that dialogue history is optional, and models

learn to detect ambiguities more effectively and efficiently with fewer preceding turns.

**Part of Speech correlations** We hypothesise that models are leveraging correlations between specific syntactic structures and disambiguation labels (e.g., increased use of anaphoric references, reduced pronoun usage, etc.). To explore this, we run a Part of Speech (POS) analysis using SpaCy (Honnibal et al., 2020).

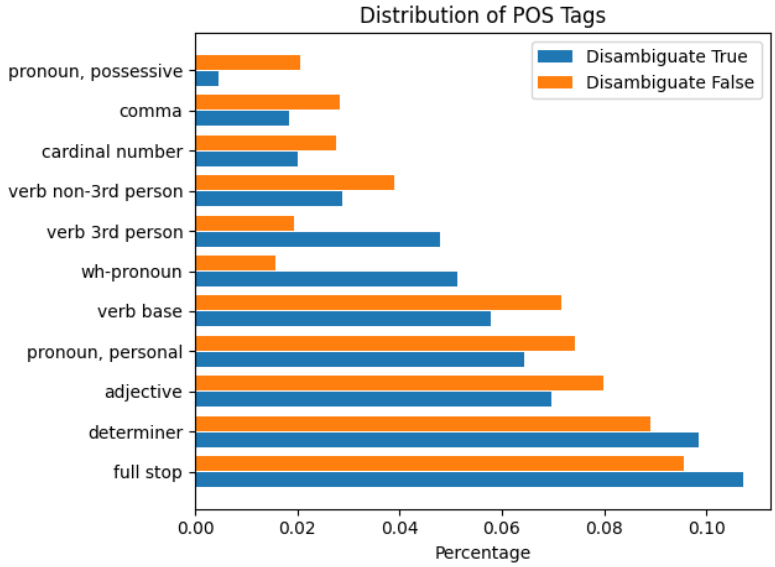



Figure 4.4: POS tags extracted from SIMMC 2.0 utterances labelled for disambiguation using SpaCy. `Disambiguate True` refers to turns where models must detect the need for disambiguation in their follow-up response, `False` otherwise.

The analysis shows that the POS distributions for utterances are similar (see Figure 4.4), although utterances with disambiguation `False` tend to have slightly more prepositions and adjectives. A Pearson coefficient reveals a weak but significant relationship between the presence of prepositions and the need for disambiguation ($r = 0.402$, $p = 9.47\text{e}{-75}$), as well as a similar trend for adjectives ($r = 0.291$, $p = 2.27\text{e}{-38}$). This suggests that utterances with more prepositional phrases (e.g., "in the top shelf") or adjectives (e.g., "light blue jacket") are less likely to require disambiguation. For example, consider the utterance: *"Can you tell me the sizes and prices of the two **grey** jeans in the **back** two cubbies on the right?"* where underline denotes prepositions and **bold** adjectives.

On the other hand, a Pearson correlation between the frequency of wh-question (e.g., *what, who*) and disambiguation shows a weak but significant relationship ($r = 0.323$, p=2.22e−47). This suggests that user utterances such as *"What's the price on those two*

*rugs?"* or *"Can you tell me who makes that sofa chair on the right?"* are more likely to require disambiguation. Additionally, we observe several other weak correlations in the data involving different syntactic tags.

When combined with the difference in mean sentence length (13.94±6.34 for utterances requiring disambiguation compared to 18.66±8.79 for those that do not), it suggests that a powerful language model may be able to exploit these patterns to accurately predict disambiguation in most cases, without relying on dialogue history or multi-modal input.

**Ambiguous coreferences** In Section 4.4, we examined the resolution of coreferences in SIMMC 2.0 dialogues. As part of the challenge, coreferences in turns where the disambiguation label was `True` (ambiguous) were excluded from evaluation, as they did not provide sufficient information for resolution. To better understand the impact of ambiguity, we tested several data subsets and found that the overall performance of our coreference resolution model (Section 4.4.2) is not significantly affected by the disambiguation label (performance remains consistent whether disambiguation is `False` or `True`).

Our analysis highlights that optimising for individual sub-tasks does not necessarily lead to the best overall system performance. For example, features such as dialogue history or visual context with LXMERT showed limited benefit with detecting ambiguities, but are still essential for the coreference task. On the other hand, linguistic cues were more effective at detecting ambiguities, which language-only models excel at.

In the turns after the system asks a clarification question (e.g., *"What price are those trousers?"* $\rightarrow$ *"Which one do you mean?"* $\rightarrow$ *"The red ones on the left"*), our model performs significantly worse at 45.36% Object F1, compared to 60.83% for the overall dataset. This decline is partly due to limitations in our model architecture, which incorporates up to two prior turns depending on token length and uses objects mentioned by the system in $turn - 1$. Clarifications requests require a more extensive dialogue history, as people often clarify elliptically, omitting previously grounded information (e.g., not repeating the word "trousers" in the previous example) (Grice, 1975). As a result, the model misses critical context. Extending the dialogue history and including objects mentioned throughout the conversation could help mitigate the performance drop. We will investigate these clarification requests further in the next chapter.

## 4.6 Discussion

In this chapter, we explored referring expressions in multi-modal dialogues as part of the SIMMC 2.0 Challenge (Kottur et al., 2021). Grounding object references in conversations is a crucial first step for effective human-agent collaboration, enabling models to develop robust representations of object properties that are essential in multi-modal contexts.

We introduced the Situated Interactive MultiModal Conversations 2.0 (SIMMC 2.0) dataset, a collection of dialogues for studying collaborative conversational AI in situated environments. Its complex visual scenes, paired with long dialogues and varied object references, provide an ideal testbed for evaluating models' abilities to resolve referring expressions and detect whether users have provided sufficient information to describe an object (ambiguous or not). We will use this dataset in the next chapter, as it features a rich mix of miscommunications within task-oriented dialogues with detailed annotations.

Our investigation focused on two key tasks from the SIMMC 2.0 Challenge: coreference resolution and ambiguity detection, both of which lie at the intersection of object grounding and miscommunication in collaborative tasks. For the coreference resolution task, we fine-tuned a custom multi-modal model that extracts visual features and object descriptions from dialogue images, and then predicts which objects a user is referring to (0, 1 or more). We framed this as a multi-label classification task and conducted an in-depth analysis of the model components to understand how it learns multi-modal representations.

The results suggest that the visual information was often too noisy and thus insufficient to robustly resolve coreferences. Specifically, we found that the visual representations tended to cluster around general object categories (e.g., jacket, sofa), failing to capture finer distinctions related to object states (e.g., hanging, folded) or abstract concepts (e.g., patterned). We attribute this limitation to a combination of factors: the visual models' pre-training, typically focused on distinguishing disparate object classes (e.g., horse, person), and the constraints of our fine-tuning process, which did not allow the model to learn more fine-grained representations as object labels were too general for this particular dataset. Ultimately, the learned representations failed to capture supplementary visual details essential for disambiguation within the cluttered SIMMC 2.0 scenes. Learning more fine-grained representations of visual objects through alternative fine-tuning strategies may help alleviating these shortcomings.

Language emerged as the most important feature, with the model primarily relying on linguistic cues to learn object representations. Interestingly, the noisy object descriptions (a textual representation of the visual RoI features), were more effective in enhancing model performance than the visual features themselves. This also suggests that the image scenes were often too cluttered with objects, or that object variations were insufficient for vision models to differentiate between them. Other experiments also showed how we can further improve model performance by allowing it to learn from special global tokens, further refining its representations.

Regarding ambiguity detection, our findings indicate that models can exploit certain biases in the SIMMC 2.0 data, such as such as the presence of adjectives and the sequence length, to classify between disambiguation labels. We show how several uni-modal models do not require dialogue history or visual context to achieve accuracies close to 90%, highlighting the importance of evaluating strong unimodal baselines on multi-modal tasks (Thomason et al., 2019).

Overall, this chapter presents experiments that evaluate multi-modality in model representations. The visual features extracted from SIMMC 2.0 scenes were too noisy for models to use effectively, leading models to rely predominantly on language. This underscores the need for improved multi-modal representations and the importance of ensuring that multi-modality is intrinsic to both the task and model. Our results in ambiguity detection also suggest that models can rely solely on language to detect ambiguous referring expressions, at least in this dataset. However, our model's coreference resolution abilities significantly deteriorate following an ambiguity and its subsequent clarification. Miscommunications are a regular part of human conversations, yet our model is unable to leverage the additional information provided by the clarification to identify target objects. To explore this further, the next chapter will examine clarifications in SIMMC 2.0 in greater depth.

# Chapter 5

# Processing Clarificational Exchanges in Multi-Modal Dialogues

This chapter explores communication breakdowns due to ambiguity and their subsequent repairs in the SIMMC 2.0 dialogues, focusing on how models include novel information in their representations. This builds upon the work from the previous Chapter 4 that briefly examined ambiguity in situated dialogues and its potential impact on Referring Expression Comprehension (REC). We leverage the SIMMC 2.0 dataset's diverse mix of ambiguities and clarifications within multi-modal dialogues to derive a taxonomy for the properties used to resolve referring expressions, such as colours or positions. With this fine-grained taxonomy, we assess how architecture and training differences in models influence their strengths and weaknesses across disambiguating object properties.

This chapter, presented at the Special Interest Group on Discourse and Dialogue (SIGDIAL'23) Conference as the paper Chiyah-Garcia et al. (2023), is organised as follows: (1) we analyse the ambiguities and clarifications within the SIMMC 2.0 dataset in Section 5.3; (2) we introduce the taxonomy and its distillation and labelling process for the dataset in Section 5.4; and (3) finally we conduct experiments with multiple models on SIMMC 2.0 clarifications using this taxonomy in Section 5.5.

## 5.1 Introduction

Human conversations rely on continuous cooperation to achieve mutual understanding (Clark, 1996; Clark and Brennan, 1991; Goodwin, 1981; Healey et al., 2018). Participants actively engage (Schlangen, 2023) to coordinate effectively (Eshghi et al., 2022; Mills, 2014), incrementally aligning their contributions to progress the interaction (Mills, 2007). We previously discussed this process in Section 2.1 as *communicative grounding*.

In this chapter, we explore Clarificational Exchanges (CEs)[1] as one of the key repair processes used to trigger negotiation and clear misunderstandings when they arise in conversation. Figure 5.1 illustrates a CE dialogue, where an initial ambiguous referring expression triggers a clarification request that is subsequently repaired.

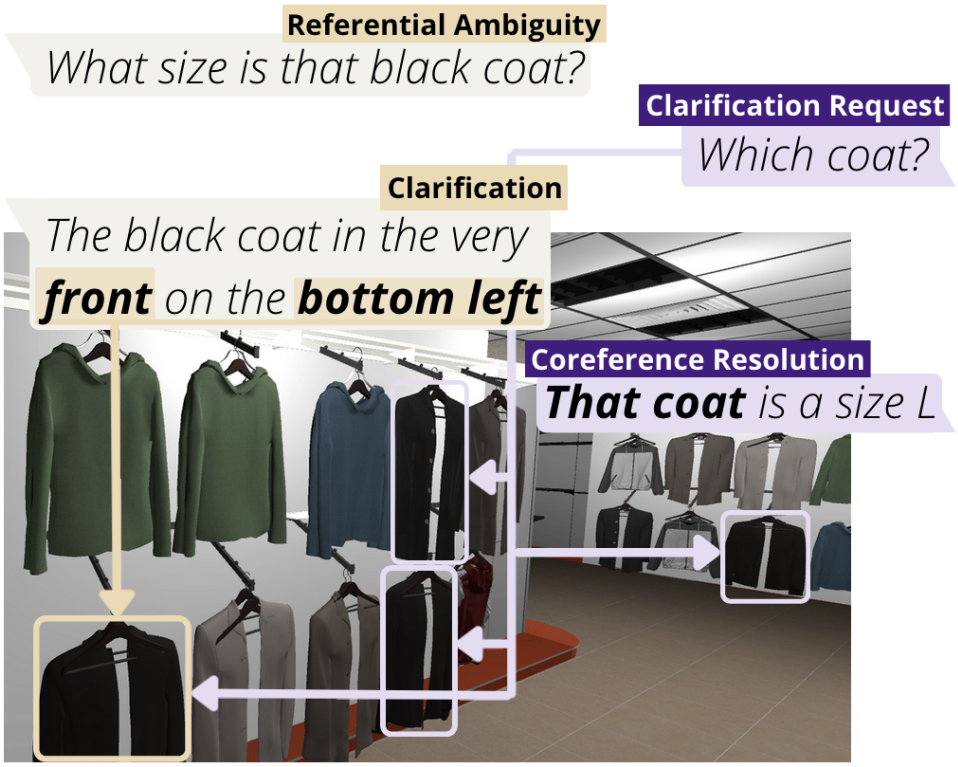



Figure 5.1: Clarificational Exchange (CE) of an ambiguous referring expression and its clarification from the SIMMC 2.0 dialogues.

CEs are a complex, context-dependent phenomena in dialogues (Purver, 2004; Purver and Ginzburg, 2004) that can occur at different levels of Clark's (1996) joint action ladder and across modalities (Benotti and Blackburn, 2021). Despite their well-acknowledged fundamental role in dialogue coordination (San-Segundo et al., 2001; Rieser and Moore, 2005; Rodríguez and Schlangen, 2004; Rieser and Lemon, 2006), CEs remain underexplored within multi-modal dialogue modelling (Benotti and Blackburn, 2021).

This chapter investigates how multi-modal models leverage CEs to resolve miscommunications in dialogues. We develop a framework to evaluate models based on the contextual information that they leverage to repair ambiguities in the SIMMC 2.0 dataset (Kottur et al., 2021, also see Section 4.3). Our focus is on ambiguous referring expressions and their associated CEs occurring at level three of Clark's (1996) action ladder:

[1] For a detailed background on CEs, see Section 2.1.4.

that of meaning and understanding (see example in Figure 5.1). To further understand models' limitations, we also propose a taxonomy that categorises the types of information required for resolution (Section 5.4.1), such as using an object's colour or position. Our findings indicate that a model's capability to process CEs for resolving referential ambiguities is closely tied to the granularity of its cross-modal representations, or how various object attributes are internally represented. Models with object-centric representations (Bengio et al., 2013; Seitzer et al., 2023) show significantly better performance processing CRs and their responses within multi-modal contexts. We identify that training models with multi-task objectives is crucial for capturing the context shifts introduced by repairs. These objectives should encourage learning disentangled object representations (Suglia et al., 2020), meaning attributes like colour or shape are represented independently (see Section 5.5.1). This capability is vital when repairs narrow the candidate pool by adding new details, such as specifying an object's colour. These are essential considerations for building models capable of addressing miscommunications in situated conversations.

## 5.2 Related Works in Clarification Processing

Clarifications are crucial for dialogue coordination, hence many works have explored when to initiate or ask a CR in discourse (Addlesee et al., 2024; Willemsen and Skantze, 2024; Willemsen et al., 2023; Madureira and Schlangen, 2024; Madureira and Schlangen, 2023; Zhu et al., 2021; Shi et al., 2022; Addlesee and Eshghi, 2021; Kuhn et al., 2023). Generating CRs has also been explored in many tasks to help models complete an overarching task, such as requesting additional information in VLN (Chi et al., 2020; Thomason et al., 2020; Nguyen et al., 2022), open-domain dialogue (Aliannejadi et al., 2021) or *when* to stop asking CRs in visual games (Shekhar et al., 2018).

Gervits et al. (2021) propose and evaluate a decision-theoretic neural network (Owen, 1978) for REC, designed to generate CRs for object attributes that minimise candidate referents. However, their work does not incorporate visual input, and their evaluation focuses on the number of turns required to resolve referents using oracle agents.

Our focus here is on assessing models' ability to effectively process and respond to clarifications, an area often treated as a secondary objective in instruction-following tasks (e.g., Padmakumar et al., 2022; Kiseleva et al., 2022; Shi et al., 2022). Multi-

modal models encode semantic knowledge about objects (Ilinykh and Dobnik, 2022; Ilinykh and Dobnik, 2021a) and spatial relations (Dobnik et al., 2018) through their learned representations. Thus, we would expect models with different input modalities (e.g., language-only vs vision-and-language) to also show variations in how they process multi-modal clarifications. Unlike previous works in clarifications, this chapter analyses the specific properties and modalities models exploit to resolve ambiguous referring expressions within situated visual dialogues.

## 5.3 Clarifications and Ambiguity in SIMMC 2.0

This section outlines specific details about CEs within the SIMMC 2.0 dataset (Kottur et al., 2021). Refer to Section 4.3 for the general dataset details and statistics.

As previously mentioned, we focus on CEs using dialogues from the SIMMC 2.0 dataset as our test data. The dataset's environments feature numerous similar-looking objects, requiring rich referring expressions to clearly identify specific objects (e.g., *"The red jacket hanging on the left wardrobe"*). The dialogues also reference multiple objects throughout their course, mixing new and previously discussed items (an average of 4.5±2.4 unique objects per dialogue), further increasing the overall ambiguity across modalities. Consequently, the dataset presents complex coordination phenomena, such as referential miscommunications and their repairs on multi-modal contexts (visual and conversational).

Although the original dataset and challenge (Kottur et al., 2021) did not explore miscommunications, two of the proposed tasks, Multi-Modal Coreference Resolution and Ambiguity Detection[2], dealt with these events indirectly: either detecting that a referring expression matched many objects and thus was ambiguous, or by resolving a coreference from an initial referring expression and subsequent repair turns. Thus, models were implicitly trained to handle dialogues that contained miscommunications as part of longer conversations and goals, unlike other datasets that solely focus on yes/no questions (i.e., GuessWhat?! (de Vries et al., 2017)) or visual ambiguities (e.g., PhotoBook (Haber et al., 2019)). The SIMMC 2.0 dataset also contains fine-grained annotations at each turn (such as objects in view, bounding boxes, referenced objects) which are ideal to explore ambiguities and repairs in dialogues.

[2]Refer to Section 4.3 for more details on the SIMMC 2.0 tasks.

### 5.3.1 CE Structure

The CEs in SIMMC 2.0 have a fixed structure, shown in Dialogue 12: (T1) an initial ambiguous *referring expression* from the user, often due to underspecified properties and multiple candidates matching the description; (T2) a system *Clarification Request (CR)* seeking additional information, which sometimes includes partial information itself (e.g., *"Which one?"* or *"The blue one?"*); and (T3) a clarification with additional object details. These CRs operate at level three of Clark's (1996) and Allwood's (1995) models of communication (meaning and understanding), and align with the *clausal* category proposed by (Purver et al., 2003; Purver, 2006; Ginzburg, 2012) or *resolving underspecification* (3b) by Schlangen (2004). We also observe that the clarification responses are typically terse (*"The black one"*), following Grice's (1975) maxim of Quantity ("give as much information as needed, and not more"), and often combine modalities (*"The black dress from earlier"*).

*(12)* Structure of a Clarificational Exchange within a SIMMC 2.0 dialogue.

| | | |
|---|---|---|
| **User:** | *I'd like the size and rating for the pants.* | **[T1: Ambiguous Utterance]** |
| **System:** | *Which pair are you asking about?* | **[T2: Clarification Request]** |
| **User:** | *I mean the gray ones in the second cabinet.* | **[T3: Clarification]** |
| **System:** | *Those are an XS, and have a 3.5 rating.* | **[T4: Resolution]** |

### 5.3.2 CE Annotation

We take advantage of the gold annotations in the SIMMC 2.0 dataset to identify dialogue turns corresponding to CEs. Turns containing ambiguous referring expressions are flagged as part of the multi-modal disambiguation task, serving as our starting point.

Our analysis focuses on understanding model capabilities both before and after the repair occurs. To achieve this, we employ the following annotation scheme:

- **Before-CR**: initial ambiguous user utterance (T2 in Dialogue 12).
- **CR**: system's clarification request (T3 in Dialogue 12).
- **After-CR**: user response that resolves the ambiguity (T4 in Dialogue 12).

In the end, we identify approximately 10% of the system turns as CRs across the `train`, `dev` and `test-dev` splits in the SIMMC 2.0 dataset.

## 5.4 Clarification Taxonomy

We explore how models leverage contextual information in CEs across modalities, extending beyond level three of Clark's joint action ladder. Previous studies have proposed more fine-grained classifications for clarification requests (see (Purver et al., 2003; Purver, 2006; Ginzburg, 2012)). Benotti and Blackburn (2021) ground clarifications into different modalities, such as socioperception (*"Are you talking to me?"*), hearing (*"What did you say?"*) or vision (*"Are you talking to me?"*), but SIMMC 2.0 clarifications are mostly on the vision/discourse modalities. The most relevant to our work is Schlangen (2004), which introduces sub-levels centred around parsing, underspecification or contextual relevance issues. However, this approach primarily addresses general sources of ambiguity (e.g., reference underspecification) and is not suitable for a more in-depth analysis of repair strategies. To address this, we propose a taxonomy based on the information or *disambiguating properties* used to resolve clarifications, emphasising the specific details provided in CRs and their responses (e.g., colours, positions).

### 5.4.1 Label Distillation and Classification

To identify the most prevalent repair strategies, we carried out a manual examination of the development subset, analysing both the system's CR and the user's clarification response turns for the information used. We categorise the observed repair strategies into the following disambiguation properties:

- **Individual Properties:** the turns include one or more references to visual or object-specific attributes (e.g., colour). We further categorise these approaches into three attributes:
  - Colours (e.g., blue, red)
  - Type (e.g., jacket, dress)
  - Property (e.g., striped, long-sleeved, hanging)
- **Relational Context:** the turns employ references to where the object is located within the environment. This can also be:
  - Spatial (e.g., leftmost, farthest)
  - Relational (e.g., next to, in the second wardrobe)
- **Dialogue History:** the turns employ linguistic cues to reference an object. These strategies use common structures such as confirming a candidate object or referring to an event in the past, which we further sub-divide into two features based on what they exploit:
  - Confirmation (e.g., *"The blue one? - Yeah"*)
  - Previous history (e.g., *"... in my cart"* or *"The jacket you mentioned earlier"*).

We left several strategies unclassified as they did not provide any meaningful additional information or are out of scope (e.g.,*"What is that lamp made of? - Sorry, which one? - I'm not sure, I think it's a lamp"*). These utterances were likely artefacts from the SIMMC 2.0 crowd-sourcing setup with ambiguous scenarios from a 'skeleton' dialogue script. As these scripts would not specify item names or attributes, crowd-workers had freedom to refer to objects openly without biasing labels (i.e., green jacket vs green t-shirt). However, this sometimes led to unnatural or nonsensical referring expressions,

which we remove from the final dataset of CEs. See Figure 5.2 for the taxonomy and example utterances.

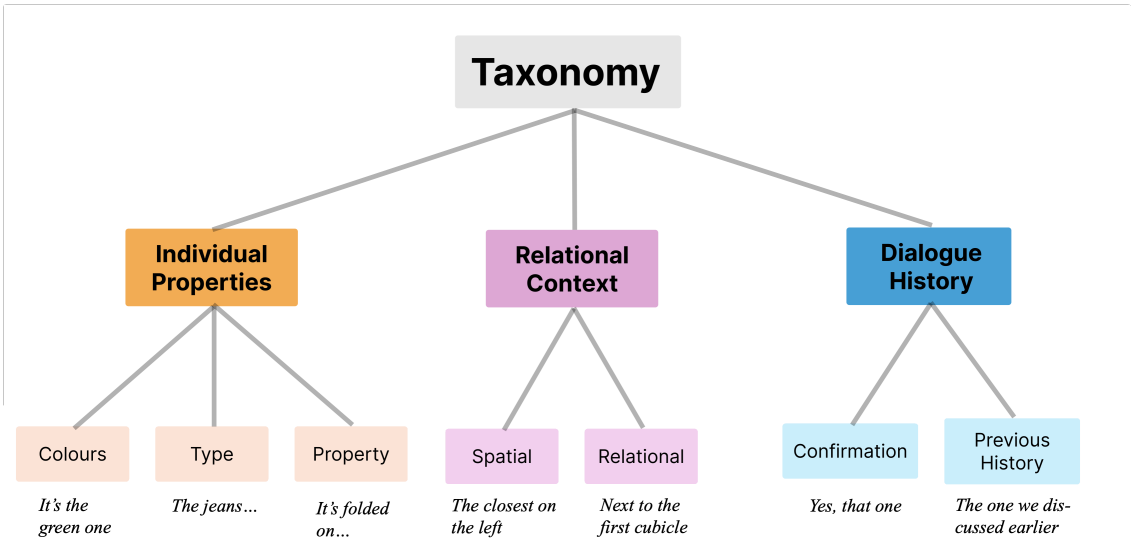



Figure 5.2: Taxonomy of repair strategies and example utterances that contain them.

**Multi-Label Classification** Many of the utterances in CEs reference multiple disambiguating properties at the same time, providing complementary information to resolve the ambiguity. For example, *"The blue jacket on the left wardrobe"* references both a colour (blue), a type (jacket) and a location (left wardrobe). To account for this, we use a multi-label classification approach to label each utterance with all the properties that it contains. Table 5.1 provides an example of how we label each utterance and the overall CE from the SIMMC 2.0 dataset using this taxonomy.

### *5.4.2 Automatic Labelling*

Following the classification outlined above, we annotate the CEs in the dataset with the disambiguating properties present. These categories are not mutually exclusive, thus we annotate utterances with all properties present in them. When multiple instances of the same property occur, we only apply the label once (i.e., if two colours are mentioned, we assign IP(Colours) only once). We annotate each turn, **Before-CR** , **CR** and **After-CR**, independently, while combining only the **CR** and **After-CR** annotations for the CE labels, as properties mentioned before the repair sequence are not considered part of the repair itself. Table 5.1 illustrates the distinction between utterance and CE labels.

| | Utterance | Labels | |
|---|---|---|---|
| | | Utterance | CE |
| **User:** | *Can you find me a black blouse?* | | |
| **System:** | *What about this black blouse in the shelf display?* | | |
| **User:** | *What's the size of the grey shirt and the grey and brown shirt?* `[Before-CR]` | IP(Colours), IP(Type) | |
| **System:** | *Which shirts are you referring to?* `[CR]` | IP(Type) | IP(Type) |
| **User:** | *I mean the grey one on the table display and the grey and brown one next to it* `[After-CR]` | IP(Colours), RC(Relational) | IP(Colours), IP(Type), RC(Relational) |

Table 5.1: Annotations of a CE with the disambiguating properties present in the utterance. Properties mentioned in the **Before-CR** turns are not part of the repair and thus we do not include them in the CE labels. Refer to Appendix A for further examples.

We begin by extracting a list of properties from the metadata, such as colours, styles and brand names. We then expand it with additional words or combinations commonly found in the data (e.g., 'lighter gray', 'hanged' instead of 'hanging'). This helps identify the specific words linked to an **Individual Property**. For the **Relational Context**, we derive a set of positional keywords (e.g., left, middle, top) and relational patterns (e.g., 'next to', 'by the corner', 'second wardrobe') from the data. Finally, **Dialogue History** is based on linguistic cues and syntactic structures (e.g., *"Okay!"*, *"... in my cart"* or *"... you mentioned"*).

To label each CE, we use regular expressions with the extracted words, patterns and synthetic structures, applying this method to both the system's CR and the user response, as the relevant information for repairing the miscommunication may be scattered across both turns (i.e., *"The black one? Yes"*).

We validated this approach by manually reviewing the development set of the SIMMC 2.0 dataset and writing unit tests to ensure that the algorithm performed as expected on the analysed CEs.

### 5.4.3 Corpus Analysis

| Split | Train | Dev | Devtest |
|---|---|---|---|
| Number of All Turns | 38127 | 3494 | 8609 |
| Number of CE Turns | 7584 | 732 | 1710 |
| Number of CEs | 3792 | 366 | 855 |
| **Disambiguating Property** | | | |
| Individual Property | 3602 | 348 | 825 |
| Colour | 3332 | 322 | 767 |
| Type | 2779 | 288 | 661 |
| Property | 1025 | 87 | 234 |
| Dialogue History | 944 | 113 | 198 |
| Previous | 569 | 74 | 123 |
| Confirmation | 455 | 44 | 93 |
| Relational Context | 3004 | 295 | 685 |
| Spatial | 2035 | 211 | 454 |
| Relational | 2521 | 250 | 570 |
| Unclassified | 7 | 0 | 1 |

Table 5.2: Taxonomy statistics within the SIMMC 2.0 Dataset using the automatic labelling and classification of the previous sections. CEs may reference several types of disambiguation properties at the same time to repair the ambiguity, hence these are not mutually exclusive. Thus, these statistics provide the frequency of different types of disambiguating properties used within CEs in the dataset.

Table 5.2 presents the descriptive statistics of the SIMMC 2.0 dataset following the automatic annotation of CEs based on the taxonomy described in Section 5.4.1. The data shows that *Individual Properties* are the most frequently used attributes for resolving ambiguities, with nearly 90% of CEs referencing a colour. The second most common attribute is object types, used in 77% of cases (e.g., jacket, t-shirt). While these visual properties are common descriptive elements (and easily identifiable in images), using them within referring expressions often leads to contextual ambiguity. Table 5.3 quantifies this under-specification, demonstrating that these properties often match with several candidate objects within the scene (i.e., sharing the same colour or type of object). Consequently, referring expressions relying on a single property are insufficient to uniquely identify the referent, so clarification responses will typically reference multiple attributes simultaneously (e.g., *"The blue jacket on the wall"*).

The unclassified CEs account for $< 0.01\%$ of the data and are removed from further analysis.

Figures 5.3 and 5.4 depict the frequency of disambiguating properties in CEs, re-

| Split | Mean Candidate Objects (SD) `Colour` | `Type` |
|---|---|---|
| All Turns | 2.58 (4.23) | 3.10 (5.16) |
| CR Turns | 4.53 (4.63) | 5.41 (5.62) |
| **Disambiguating Property** | | |
| Individual Property | 4.57 (4.77) | 5.52 (5.69) |
| Dialogue History | 4.95 (5.25) | 4.57 (4.82) |
| Relational Context | 4.63 (4.66) | 5.78 (5.91) |

Table 5.3: Statistics on the contextual ambiguity of referring expressions within the SIMMC 2.0 dataset. This analysis measures the number of objects in the scene/context that match the referenced object **Type** (*jacket, t-shirt...*) or **Colour** (*blue, yellow...*) in particular splits (i.e., All Turns, or only turns with expressions referencing the Relational Context). Ambiguities frequently arise because multiple objects share properties commonly used, so these properties alone are not enough to uniquely identify an object in most cases. For instance, in CR Turns, a simple reference like *"The red one"* matches an average of 4.53 potential objects in the context.

vealing that the most common approach for resolving ambiguities is a combination of *Individual Properties* and *Relational Context*. On average, each CEs references a mean of 3.37 attributes (median: 3, standard deviation: 1.13), with some instances reaching up to 7 attributes (see Dialogue 13 below). Figure 5.5 shows the evolving nature of CEs, where initial references mostly use Individual Properties and visual attributes (type, colour) and incorporate Referential Context towards their resolution. The distribution of disambiguating attributes aligns with Dale and Reiter's (1995) proposed attribute ranking for generating referring expressions.

*(13)* **User:** *What brand is the black coat on the rack by the door, behind the grey coat you recommended?*

**System:** *I am not sure what coat you are referring to, could you explain?*

**User:** *I mean the black coat that hangs second from us on the left side of the floor hanger by the entrance. I also want to know the brand of the coat that hangs first on the left of that hanger.*

**System:** *Thank you! The jacket that is closest to us on that side of the hanger is by Art News Today. The jacket behind it is by Cats Are Great.*

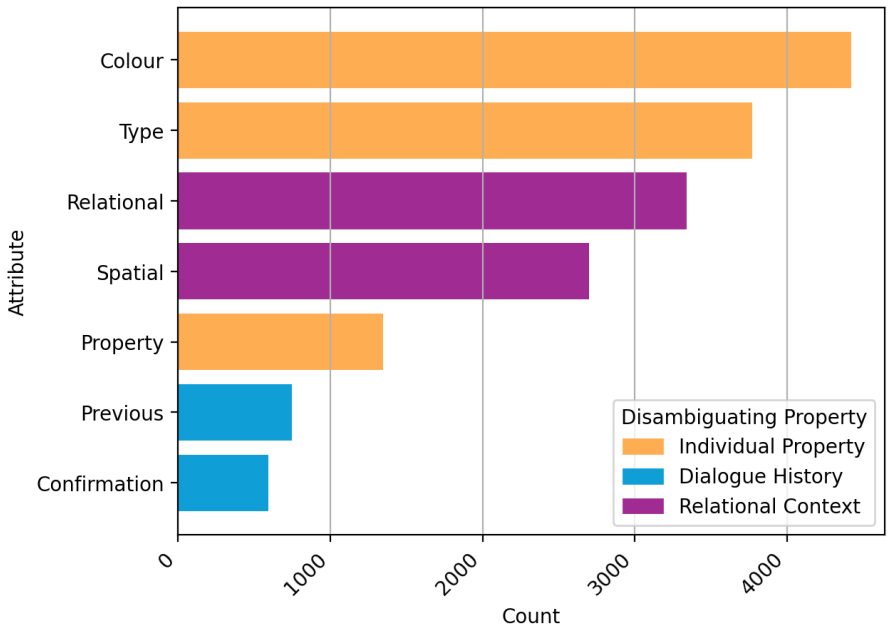



Figure 5.3: Frequency of disambiguating attributes found in CEs for the SIMMC 2.0 dataset. The distribution aligns with the preferred attribute ordering proposed by Dale and Reiter (1995) for generating referring expressions.

## 5.5 Experimental Evaluation

This section evaluates how different model architectures and training objectives impact downstream task performance, particularly when repairing ambiguities in SIMMC 2.0 dialogues.

### *5.5.1 Methodology*

#### Metrics

Following the previous chapter and Kottur et al. (2021), we evaluate coreference resolution performance with **Object F1**, see Equation (5.1). Refer to Section 4.4.3 for further details. We use this metric as a proxy of how well models process clarifications as part of the overall coreference resolution task.

$$Object\ F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.1}$$

In addition to Object F1, we analyse the difference in performance between user turns before and after a clarification, referred to as **Before-CR** and **After-CR** respectively (§5.3.2). Intuitively, a model that effectively handles clarifications should show

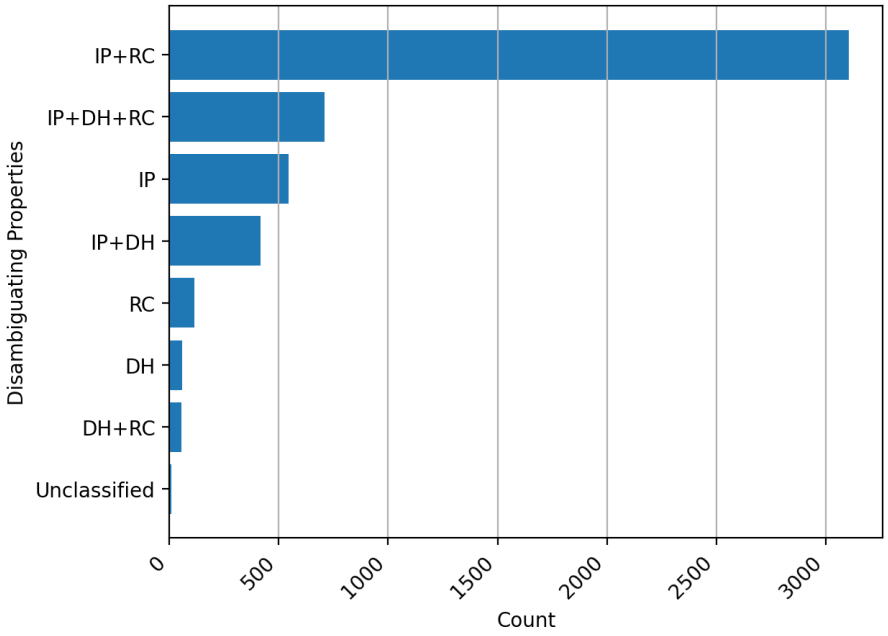



Figure 5.4: Frequency of combined disambiguating properties in CEs for the SIMMC 2.0 dataset. Most CEs use a mix of Individual Properties (IP) and Relational Context (RC) to resolve ambiguities, followed by a combination of all three groups including Dialogue History (DH).

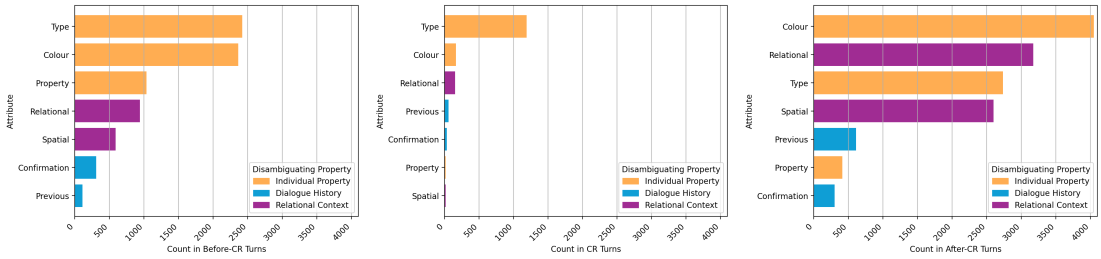



Figure 5.5: Changes in frequency of disambiguation attributes before a CR, in a CR and after the CR.

improved performance after a clarification, reflected by a higher Object F1 in the set of **After-CR** turns. On the other hand, poorer performance in **Before-CR** turns may signal confusion or uncertainty in general. We quantify this difference as the **Relative Delta** $\Delta$ (see Equation (5.2)), which allows us to compare performance across models handling clarifications.

$$\textit{Relative Delta } \Delta = \frac{\text{Object F1}_{\text{After-CR}} - \text{Object F1}_{\text{Before-CR}}}{\text{Object F1}_{\text{Before-CR}}} \tag{5.2}$$

### Models

For our evaluation, we selected publicly available state-of-the-art pre-trained models that participated in the SIMMC 2.0 challenge[3]. We provide key model details and differences below (also see Figure 5.6), refer to the original papers for more comprehensive architectural information.

**Baseline$_{\text{GPT-2}}$** The baseline model included in the initial release of the SIMMC 2.0 dataset (Kottur et al., 2021) is based on GPT-2 (Radford et al., 2019) and is trained with causal language modelling to generate the next token in a sequence. This model approaches the task as a language-only generation task, predicting object tokens that correspond to the referenced objects in each dialogue turn (e.g., OBJ_12), see Figure 5.6a. It is jointly trained alongside the other challenge objectives – coreference resolution, dialogue state tracking, and response generation– achieving an average coreference resolution Object F1 score of 36.6%.

**GroundedLan$_{\text{GPT-2}}$** This model from (Hemanthage and Lemon, 2022) improves upon the baseline model with additional training to ground multi-modal attributes to object tokens, see Figure 5.6b. It is also GPT-2-based and trained with all challenge goals to generate the next language token, achieving an F1 score of 67.8% in coreference resolution.

**VisLan$_{\text{LXMERT}}$** We take the LXMERT-based (Tan and Bansal, 2019) model previously described in Section 4.4.2. This model integrates visual information from the visual scenes with dialogue to predict referenced objects at each turn through multi-modal cross-attention modules, see Figure 5.6c. It first extracts object attributes to create textual descriptions and incorporates visual feature embeddings encoded by a Detectron2 model (Wu et al., 2019). The model then outputs a probability for each object in the scene, indicating the likelihood that it is being referenced in a specific dialogue turn. The final selected objects are those that exceed a predetermined threshold. This model is only trained for multi-modal coreference resolution, achieving an Object F1 score of 68.6%.

[3]Some models were not publicly available, and others had missing code or weights to reproduce their results.

**MultiTask$_{\text{BART}}$** We use the model developed by the winning team of the coreference challenge (Lee et al., 2022), which is based on BART (Lewis et al., 2020) and achieved an Object F1 score of 74%. This model leverages a pre-trained ResNet model (He et al., 2016) to encode each object along with its non-visual attributes using learnable embeddings that are subsequently mapped to match BART's dimensionality. The object location is also encoded using bounding box information and a location embedding layer. Finally, canonical object IDs are employed to ground relationships between object locations and both visual and non-visual attributes, see Figure 5.6d.

The model is jointly trained across all challenge tasks and several additional secondary objectives designed to learn disentangled object representations (Bengio et al., 2013) through object-attribute slot prediction for each scene object.

Specifically, **disentangled object representations** mean that attributes such as colour, type or material are encoded in a way that allows the model to reason about each property independently. They aim to capture distinct factors of variation in the data separately (e.g., `[colour=red,type=shirt]`), so that modifying the representation of one attribute (e.g., changing colour from red to blue) ideally does not affect the representation of other attributes (e.g., type remains constant as in `[colour=blue,type=shirt]`). Achieving this disentanglement often involves training objectives, like the mentioned slot prediction, that encourage independence between different parts of the learned representation, potentially leading to better generalisation and robustness when specific attributes are referenced or newly introduced, as often happens in CEs.

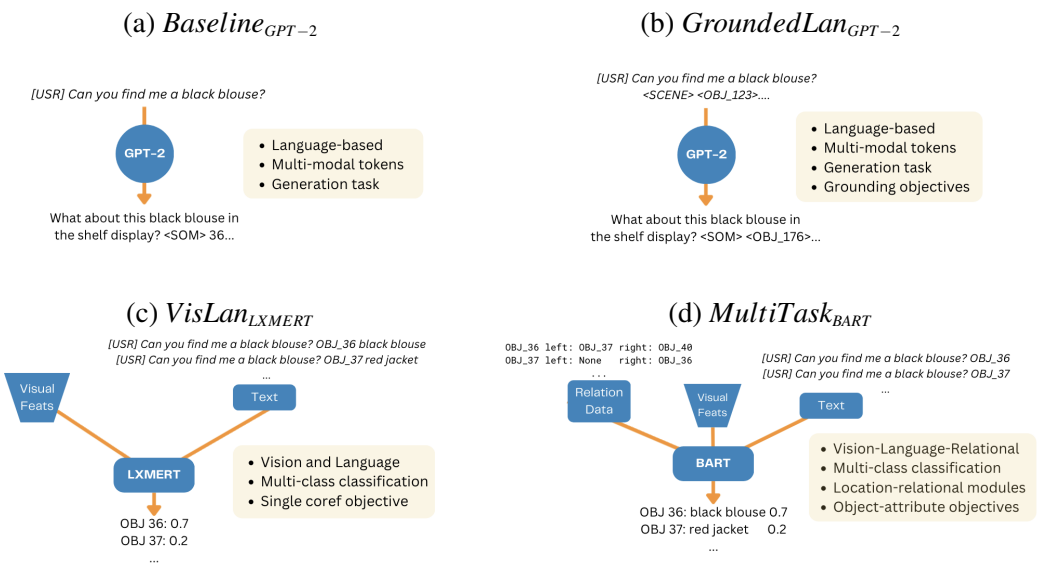



Figure 5.6: Summary of the models' architectures and input/outputs.

#### Training and Evaluation

We fine-tune each model using the default hyperparameters specified in their respective implementations with the `train` split of the SIMMC 2.0 dataset. As the gold data from the `test-std` split is unavailable[4], we conduct our evaluations using the `devtest` data. We successfully reproduced the Object F1 scores reported in the original model papers within a few decimal points. The remainder of this section presents results and discussions based on the `devtest` split.

### *5.5.2 Results*

We use the coreference resolution task from the SIMMC 2.0 dataset as our testbed to evaluate models' ability to handle multi-modal referential CEs and resolve ambiguous referential descriptions. Since CEs naturally occur in SIMMC 2.0 dialogues, we expect that models will learn to address these repairs as part of the overall task. While **After-CR** turns often contain more descriptive attributes (as shown in Figure 5.3), we only evaluate models' ability to correctly identify the objects, rather than the presence of additional information or referring expressions, which could be noisy or not useful to repair the ambiguity (e.g., repeating old with new object properties). This section presents the results of training and evaluating models on multi-modal coreference resolution, with ablations on data subsets based on the different categories of disambiguating properties discussed in Section 5.4.1. Table 5.4 presents the main experimental results of this section.

#### Referential Ambiguities

First, we investigate whether referential ambiguities pose a challenge for the models and whether clarifications are generally necessary. The upper section of Table 5.4 shows that all models – except $Baseline_{GPT-2}$ – perform worse in **Before-CR** turns compared to **All Turns**. This suggests that these turns lack sufficient information to uniquely identify the referent objects, leading to referential ambiguities and a lower Object F1 for models.

Secondly, we observe that F1 scores are generally higher in **After-CR** compared to **Before-CR** turns across all models but $Baseline_{GPT-2}$ , indicating that most models can leverage clarifications to some extent, as shown in Figure 5.7. Notably, the $VisLan_{LXMERT}$

[4]This data was not publicly accessible during or after the challenge.

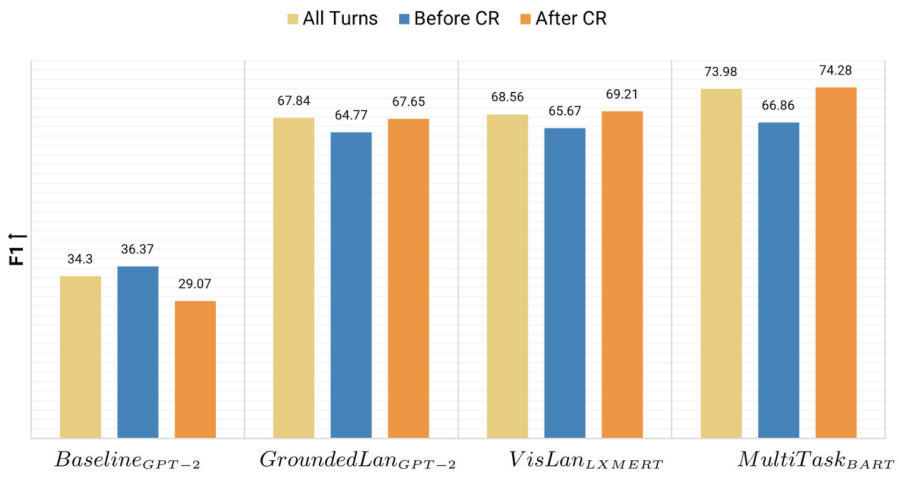



Figure 5.7: Model performance in resolving coreferences across all dialogue turns (**All Turns**), ambiguous user turns (**Before-CR** ) and user clarification turns (**After-CR** ).

and *MultiTask*$_{BART}$ models even outperform their **All Turns** results in **After-CR** turns, likely due to the additional information provided during clarifications, which aids in resolving coreferences.

The unexpectedly high scores for the *Baseline*$_{GPT-2}$ model in **Before-CR** turns, and the low scores in **After-CR** turns, suggest that this model may be relying on linguistic phenomena and effectively using previously mentioned objects and their canonical IDs, proposed in Chiyah-Garcia et al. (2022) and discussed in Chapter 4. However, the model's performance drops significantly when it is essential to carry over cross-turn information and ground it within the dialogue, which is necessary for **After-CR** turns.

### Disambiguating Properties

We use the taxonomy of repair strategies from Section 5.4.1 to ablate data subsets and probe how models handle different information in CEs. Figure 5.8 summarises model performance for the distinct disambiguating properties.

**Individual Property** All models except *Baseline*$_{GPT-2}$ demonstrate comparable performance in **Before-CR** turns that reference individual properties. *GroundedLan*$_{GPT-2}$ and *VisLan*$_{LXMERT}$ moderately improve their Object F1 scores in the subsequent **After-CR** turns, while *MultiTask*$_{BART}$ achieves the most significant improvement (+11.3% Δ). Individual object properties in this dataset often relate to concepts in the visual context, which may be challenging to understand beyond simple attributes like colours (e.g., long-sleeved or folded). As discussed in Section 5.4.3, users frequently defaulted to these

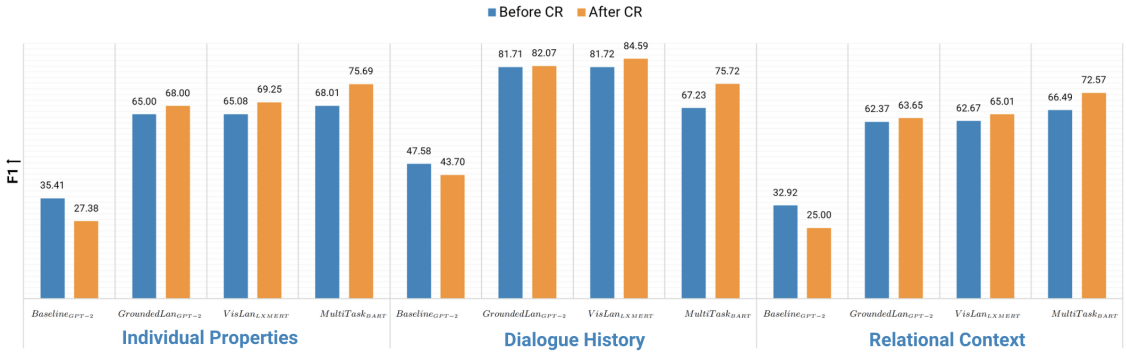



Figure 5.8: Model performance in resolving coreferences across disambiguating properties for ambiguous user turns (**Before-CR** ) and user clarification turns (**After-CR** ).

attributes for both referring expressions or clarifications.

The $GroundedLan_{GPT-2}$ model implicitly encodes object attributes using a global object ID, enabling the model to learn latent information during training that carries over to evaluation sets (i.e. `<OBJ_256>`). On the other hand, the $VisLan_{LXMERT}$ model explicitly encodes colours and object types through textual descriptions (i.e. `blue hoodie`) and implicitly via the visual region of interest features. This approach explains the slightly higher performance of $VisLan_{LXMERT}$ for these attributes. However, the vision module of $VisLan_{LXMERT}$ is not explicitly trained to detect complex or abstract properties. It primarily focuses on attributes such as colours or shapes (i.e., blue hoodie), relying on visual features to represent more nuanced information (i.e., folded or hanging). This limitation may affect its ability to fully capture more intricate object properties in the coreference resolution task.

The multi-task learning objectives employed by the $MultiTask_{BART}$ model during training enable the model to obtain more fine-grained and disentangled representations compared to approaches that rely solely on vision. These enhanced representations significantly aid in resolving ambiguities related to individual object properties. Previous works (Suglia et al., 2020) suggest that leveraging explicit object attributes can effectively reduce the number of potential referents in a given context, leading to less ambiguity and thus improved performance. By explicitly modelling these attributes, $MultiTask_{BART}$ can more accurately disambiguate references and resolve complex coreferences.

**Dialogue History** The $GroundedLan_{GPT-2}$ and $VisLan_{LXMERT}$ models perform well when the CEs reference the dialogue context. Their initial Object F1 scores exceed the mean model performance for All Turns (>73.9%), demonstrating their strong capability to carry

information across turns and suggesting that CRs may not be necessary in these instances. $VisLan_{LXMERT}$ further leverages this information in **After-CR** turns, improving its performance to 77.4% (+4.7%) F1 and achieving the highest score for these properties, whereas $GroundedLan_{GPT-2}$ exhibits a slight performance decline. In contrast, $MultiTask_{BART}$ improves its performance to 72.5% F1 (+11.6% $\Delta$) but does not exhibit the same proficiency to leverage the linguistic context as the other models in the first place (64.9% F1 in **Before-CR** turns). This likely stems from its multi-task training, which involves specific loss functions focused primarily on visual and relational information. As a result, while the model develops robust visual and relational object representations, it compromises the quality of BART's pre-trained language representations.

**Relational Context** Relational clarifications seem to be the most challenging referring strategy for models, yielding the lowest Object F1 scores overall. However, the $MultiTask_{BART}$ model handles this information considerably better than the others, improving by $\Delta$ +9.4% to 71.8% F1. This strength highlights the model's ability to not only encode visual attributes of objects but also capture the relationships between them within the scene more effectively. For instance, this model is able to capture object positions in the scene and how they relate to each other. While the $VisLan_{LXMERT}$ model also encodes positional information such as bounding box coordinates, it struggles to learn and leverage these features, discussed in Chapter 4 and Chiyah-Garcia et al. (2022). This limitation aligns with previous findings, which demonstrate that multi-modal models often struggle with spatial concepts like position (Ilinykh and Dobnik, 2021b; Ilinykh and Dobnik, 2022; Pantazopoulos et al., 2024) and tend to rely on language biases instead (Salin et al., 2022).

### Model Proficiencies

Our results suggest that the choice of model architecture and training objectives significantly influence how models handle different disambiguating properties. We summarise these insights as follows:

- **Individual Properties**: models that leverage explicit object attributes outperform those that do not (e.g., $MultiTask_{BART}$ vs $GroundedLan_{GPT-2}$ ). Training with explicit object attributes facilitates fine-grained, disentangled object representations

crucial for repairing ambiguities, such as the textual descriptions from $VisLan_{LXMERT}$ or the additional slot-prediction objective from $MultiTask_{BART}$ . When using vision, models should also train to detect complex or abstract properties (e.g., folded or hanging) as with $MultiTask_{BART}$ to maximise the effectiveness and robustness of the model's visual representations.

- **Dialogue History**: models with robust language modelling capabilities excel at carrying information across turns compared to others (e.g., $GroundedLan_{GPT-2}$ and $VisLan_{LXMERT}$ vs $MultiTask_{BART}$ ). Their superior linguistic capability allows them to effectively leverage dialogue history to track mentioned objects (as shown by their significant above-average performance in **Before-CR** turns). Models fine-tuned with multi-modal objectives should balance and maintain their pre-trained language capabilities, for example by training with objectives that emphasise information carry-over across dialogue turns.

- **Relational Context**: models trained with specific relational objectives demonstrate better performance in processing spatial relationships compared to alternatives (e.g., $MultiTask_{BART}$ vs $GroundedLan_{GPT-2}$ ). These objectives enable models to encode object positions and inter-object relationships, which are essential for distinguishing similar objects in different locations (e.g., *"The jeans on the left"*). Visual features or bounding boxes alone were insufficient to learn this relational knowledge (e.g., as with $VisLan_{LXMERT}$ ). Additionally, we should design training frameworks that prevent models from relying on language biases instead of true spatial understanding (e.g., $Baseline_{GPT-2}$ ).

## 5.6 Discussion

This chapter examined the mechanisms by which models process and resolve multi-modal referential ambiguities using the SIMMC 2.0 dataset (Kottur et al., 2021). Ambiguities frequently occur in situated human conversations when interactants cannot fully comprehend or identify a referred object or action. In these cases, conversational participants must engage in Clarificational Exchanges CEs to repair miscommunications and achieve mutual understanding.

We first investigated the most frequent strategies to repair ambiguities in the SIMMC

2.0 dialogues. From this analysis, we distilled a taxonomy of CEs based on the specific disambiguating properties used to resolve each type of ambiguity. We then evaluate several state-of-the-art models, each with distinct architectures and training regimes, on the task of coreference resolution and across data subsets derived from the taxonomy. This evaluation assessed the extent to which these models can effectively process and integrate repairs in dialogues, and to understand how this capability functions within the broader context of multi-modal coreference resolution tasks.

Our findings reveal that while language-based models demonstrate reasonable performance and can develop some multi-modal understanding, they often struggle to fully leverage novel information provided through clarifications. Incorporating vision components into models is an important but not strictly necessary step to handle multi-modal scenarios, as it improves the processing of multi-modal referential expressions and their repairs. Our model, $VisLan_{LXMERT}$ , performs well across both modalities and effectively carries information across turns, particularly excelling with CEs related to dialogue history due to its strong dialogue context modelling. However, the visual attributes encoded by the model do not sufficiently capture spatial locations or object relations within the scene, an issue with vision and language models (Ilinykh and Dobnik, 2021b; Ilinykh and Dobnik, 2022; Pantazopoulos et al., 2024).

Incorporating additional learning objectives, especially those aimed at learning disentangled object representations (Bengio et al., 2013), and encoding relationships between objects and their location, yields more robust models for handling CEs in multi-modal situated dialogues. These additions can however undermine the strong language context provided by pre-trained language models like BERT (Devlin et al., 2019) or BART (Lewis et al., 2020), diminishing the context carry-over across dialogue turns.

To build models that can effectively resolve referential ambiguities in situated dialogues, these results suggest that we need *disentangled object-centric representations* that encode information about both attributes and relationships (Seitzer et al., 2023), and which can *dynamically* adapt based on information exchanges within in the dialogue context. This balance between language objectives and explicit object-attribute modelling demonstrates a promising approach for handling miscommunication in multi-modal environments, particularly in contexts where visual, spatial and linguistic information must be integrated effectively. In the next chapter, we explore training models to handle miscommunications, focusing on learning to exploit relational context in ambiguous dialogues.

| Model | $Baseline_{GPT-2}$ | | | $GroundedLan_{GPT-2}$ | | | $VisLan_{LXMERT}$ | | | $MultiTask_{BART}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Split** | **Before-CR** | **After-CR** | **Δ %** | **Before-CR** | **After-CR** | **Δ %** | **Before-CR** | **After-CR** | **Δ %** | **Before-CR** | **After-CR** | **Δ %** |
| **All Turns** | 34.1 (.01) | | | 67.6 (.01) | | | 68.3 (.01) | | | 73.8 (.01) | | |
| **CE Turns** | 36.4 (.01) | 29.1 (.01) | -20.1 | 64.8 (.01) | 67.7 (.01) | +4.4 | 65.7 (.01) | 69.2 (.01) | +5.4 | 66.9 (.01) | 74.3 (.01) | +11.1 |
| *Disambiguating Property* | | | | | | | | | | | | |
| **Individual Property** | 35.5 (.01) | 28.1 (.01) | -20.8 | 64.5 (.01) | 67.5 (.01) | +4.7 | 65.2 (.01) | 69.0 (.01) | +5.8 | 67.0 (.01) | **74.5** (.01) | +11.3 |
| Colour | 34.7 (.01) | 26.6 (.01) | -23.4 | 64.4 (.02) | 67.3 (.01) | +4.5 | 64.6 (.01) | 68.4 (.01) | +5.9 | 67.9 (.02) | **75.2** (.01) | +10.7 |
| Type | 31.3 (.02) | 24.8 (.01) | -20.8 | 62.7 (.02) | 66.2 (.02) | +5.6 | 63.7 (.02) | 67.0 (.02) | +5.1 | 67.2 (.02) | **75.0** (.01) | +11.6 |
| Property | 31.7 (.03) | 23.3 (.02) | -26.6 | 61.6 (.03) | 67.0 (.03) | +8.7 | 66.9 (.03) | 67.5 (.03) | +1.0 | 69.0 (.03) | **72.3** (.03) | +4.8 |
| **Dialogue History** | 42.9 (.03) | 36.0 (.03) | -16.0 | 74.8 (.03) | 74.6 (.03) | -0.3 | 73.9 (.03) | **77.4** (.02) | +4.7 | 64.9 (.03) | 72.5 (.03) | +11.6 |
| Previous | 42.1 (.04) | 38.8 (.04) | -7.9 | 77.8 (.03) | 76.4 (.03) | -1.9 | 78.1 (.03) | **81.6** (.03) | +4.5 | 65.1 (.04) | 73.8 (.04) | +13.3 |
| Confirmation | 44.9 (.04) | 35.0 (.04) | -22.1 | 72.4 (.04) | 72.7 (.04) | +0.4 | 71.5 (.04) | 73.8 (.04) | +3.2 | 67.2 (.04) | **74.3** (.04) | +10.6 |
| **Relational Context** | 31.8 (.01) | 25.1 (.01) | -21.0 | 62.2 (.02) | 63.9 (.02) | +2.7 | 62.8 (.02) | 64.8 (.02) | +3.1 | 65.7 (.02) | **71.8** (.02) | +9.4 |
| Spatial | 32.0 (.02) | 24.9 (.02) | -22.1 | 60.3 (.02) | 60.8 (.02) | +1.0 | 63.8 (.02) | 64.2 (.02) | +0.6 | 64.4 (.02) | **69.7** (.02) | +8.3 |
| Relational | 30.4 (.02) | 24.5 (.02) | -19.5 | 62.5 (.02) | 65.1 (.02) | +4.2 | 60.7 (.02) | 62.7 (.02) | +3.3 | 65.6 (.02) | **71.7** (.02) | +9.2 |

Table 5.4: Results for the coreference resolution evaluation in SIMMC 2.0, presented in two sections. The upper section shows model performance at resolving coreferences for all dialogue turns (All Turns) or only those involving CEs (CE Turns). The lower section reports model performance on ablated data subsets, categorised by the disambiguating property used in the CEs. Performance is measured using **Object F1** (higher is better ↑) (SD) and **Relative Delta** Δ (positive Δ show improvements over **Before-CR** turns). Bold numbers represent the top-performing models for a particular disambiguating property.

# Chapter 6

# The BlockWorld-Repairs Benchmark for Handling Multi-Modal Corrections

This chapter investigates the ability of vision and language models to generalise in interpreting and responding to repairs in situated dialogues. Building on insights from Chapters 4 and 5, we move beyond simple object-detection tasks to explore coordination with large VLMs. Using the collected corpus from Chapter 3, we train models and experiment with specialised loss functions to encourage models to learn from interactive settings. To gain further insights into the way humans handle repairs in a highly ambiguous tabletop manipulation task, we complement the corpus with a human study that provides baseline data and annotations, enabling detailed comparisons between human and model performance.

We presented this research at the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP) as the paper Chiyah-Garcia et al. (2024b). This chapter is structured into three main sections: (1) formalisation of the BlockWorld-Repair (BW-R) dataset and benchmark; (2) experimental evaluation of visual-and-language models using various loss criteria; and (3) a comprehensive error analysis and comparison between human and model performance.

## 6.1 Introduction

The way NLU is often described within the Natural Language Processing (NLP) literature is as a unilateral, passive process (see especially Wang et al., 2018; Wang et al., 2019) whereas it is an (inter)active one (see Schlangen, 2023, for expansive discussion) where people work together to negotiate shared understanding. Thus, the ability to interpret and generate effective repair sequences is essential to *robust* and *faithful* Conversational AI

technology. This is particularly important in settings requiring more fine-grained levels of mutual understanding, such as in embodied human-machine collaboration (Suglia et al., 2024).

In particular, this chapter focuses on Third Position Repairs (henceforth TPR; Schegloff, 1992, discussed in Section 2.1.4) in multi-modal environments. TPRs occur when the addressee initially misunderstands the speaker (see Figure 6.1 at T1, the *trouble source* turn), responds based on this misunderstanding (at T2), which then reveals the misunderstanding to the addressee who then goes on to correct the misunderstanding (at T3). Table 6.1 provides a comparison between the type of repairs explored in this chapter, TPRs, and those investigated in Chapter 5, CEs. The fundamental difference is in who initiates the repair: with TPRs, speakers *self-initiate* repairs upon recognising a misunderstanding in the addressee's response, whereas with CEs, addressees explicitly request a clarification as they realise a lack of information (*other-initiated*). Despite both phenomena being prevalent in dialogue, TPRs may potentially be more difficult to detect and interpret due to the absence of an explicit repair initiation (i.e., a Clarification Request)). Instead, speakers typically recognise the need for repair through a perceived mismatch between the addressee's actual and their anticipated response.

| | **Clarificational Exchange** | **Third Position Repair** |
|---|---|---|
| Initiator | Other-initiated | Self-initiated |
| Repair | Self-repair | Self-repair |
| Chapter | Chapter 5 | Chapter 6 (here) |
| *Example* | | |
| T1 | User: I'd like the size and rating for the pants. | User: Take the third most left block and move it slightly to the right underneath the adjacent block. |
| T2 | System: Which pair are you asking about? | System: Do you mean this block? [Bounding Box] |
| T3 | User: I mean the gray ones in the second cabinet. | User: No, the block below that. |
| T4 | System: Those are an XS, and have a 3.5 rating. | System: Got it [Bounding Box] |

Table 6.1: Comparison of the types of repairs discussed in this thesis, TPRs and CEs, with examples. In CEs, the addressee requests a clarification in T2, whereas in TPRs, the speaker realises there was a misunderstanding based on the addressee's response in T2.

TPRs have been largely neglected in the NLP community, likely due to underestima-

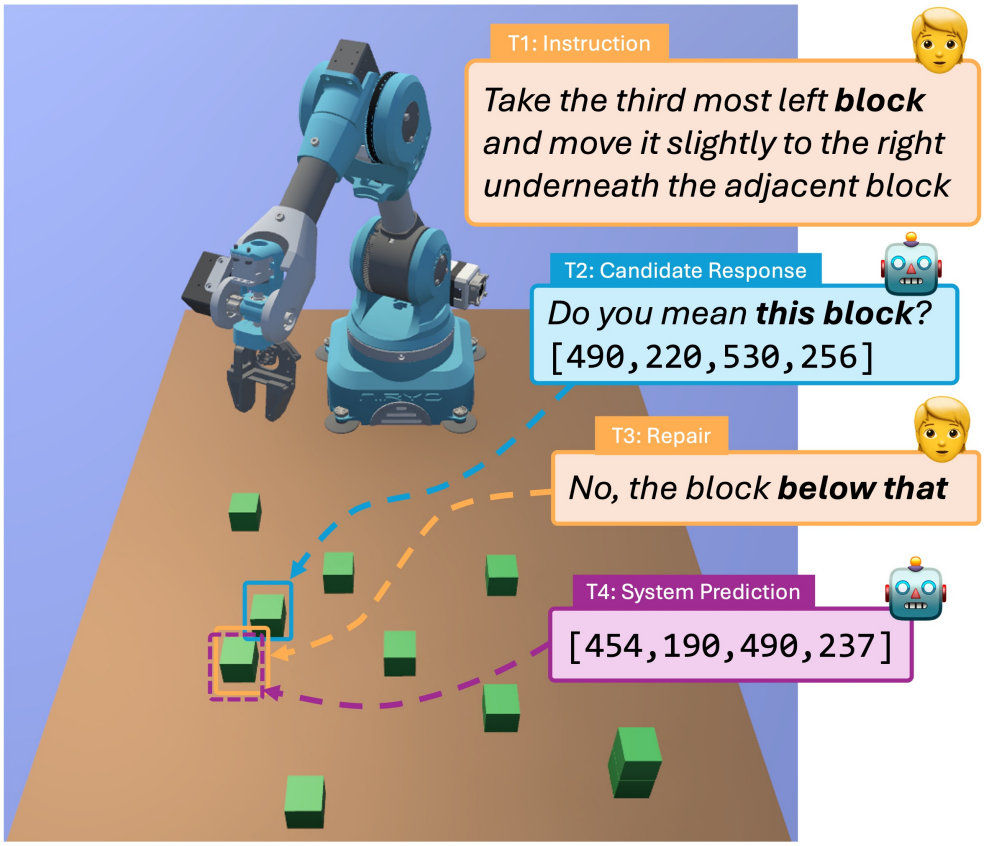



Figure 6.1: Dialogue with a Third Position Repair, in which the agent must accurately interpret the repair to generate the correct bounding box prediction.

tion of the importance of repair phenomena in dialogue, and thereby lack of appropriate data and frameworks for model training and evaluation. In this chapter, we show that this is also consequential for how language models should be trained to process TPRs including the specific objective functions involved.

This chapter explores TPRs with the corpus from Chapter 3 of collaborative dialogues in a tabletop manipulation task in the Blocks World. We introduce the BlockWorld-Repairs benchmark, an extension of the dataset presented in Chapter 3 with ambiguous referential instructions and their repairs, which includes human annotations. To validate the dialogue quality and establish a human baseline for the task from Section 3.4, we conduct a study in which participants attempt to complete the task using these dialogues, playing the role of the agent.

We then experiment with state-of-the-art VLMs, demonstrating that they can learn to process TPR sequences through specialised training regimes despite initially struggling out of the box. Leveraging insights from Chapter 4 and Chapter 5, we steer model training to focus on relevant tokens in the input, resulting in better performance and the

ability to generalise towards handling both unseen instructions and TPRs. Despite these improvements, a significant performance gap remains between VLMs and humans. To better understand this discrepancy, we carry out an in-depth error analysis, revealing that models often struggle with references that humans easily resolve, and vice versa.

## 6.2 Related Works on Interpreting Repairs

**Computational Models of Repair** Considerable attention has been paid to the interpretation and generation of *self-repairs*[1] within the same conversational turn, e.g. *"User: I want to go to London uhm sorry Paris"* (see Hough and Schlangen, 2015; Hough, 2015; Shalyminov et al., 2017; Eshghi and Ashrafzadeh, 2023; Buß and Schlangen, 2011; Hough and Purver, 2012, among others). Similarly, previous works have explored CRs (e.g. *"Pardon/what/who?"*) in conversational models (San-Segundo et al., 2001; Purver, 2004; Purver and Ginzburg, 2004; Rieser and Moore, 2005; Rodríguez and Schlangen, 2004; Rieser and Lemon, 2006), including when to pose a CR (Addlesee et al., 2024; Madureira and Schlangen, 2024; Madureira and Schlangen, 2023; Zhu et al., 2021; Shi et al., 2022; Addlesee and Eshghi, 2021), but existing systems either remain limited (e.g., Curry et al., 2018) or do not support this at all[2]. Few works evaluate the ability of models to process their responses (Gervits et al., 2021; Aliannejadi et al., 2021) as we do here.

In particular, Balaraman et al. (2023) provides the first large dataset of Third Position Repairs to assess LLMs. However, their focus is on unimodal repairs (language-only) in the context of Conversational Question Answering, and do not train any of the evaluated models.

**VLMs for Collaborative Tasks** There have been many attempts to derive Vision + Language models that use pre-trained visual encoders to solve vision-language tasks (e.g., (Liu et al., 2024b; Laurençon et al., 2024), *inter alia*). It is important to note that most of these models are trained to follow instructions that can be specified in a single turn. Many tasks can be included in this category such as Visual Question Answering (Antol et al., 2015), Image Captioning (Lin et al., 2014b) and Referring Expression Comprehension (Yu et al., 2016a). To solve these tasks, models are trained to maximise the

[1] See previous discussion in Section 2.1.4.
[2] See Purver et al. (2018) for an overview.

likelihood of the responses in the training data using supervised learning methods or using reinforcement learning from human feedback (Ouyang et al., 2022), ignoring the fact that a conversation is a process that unfolds over multiple turns where each of which matters to make correct decisions. For instance, in tasks such as Visual Dialogue (Das et al., 2017), it is well-known that a very limited number of responses are dependent on the dialogue history (Agarwal et al., 2020), making them not a suitable benchmark for truly collaborative tasks requiring the ability to establish common ground.

As highlighted by Suglia et al. (2024), many Embodied AI benchmarks (e.g., ALFRED (Shridhar et al., 2020), Simbot Arena (Gao et al., 2023b), etc.) also have similar issues considering that the level of ambiguity is reduced to the minimum, minimising the need for any form of correction. For this reason, BW-R represents the first benchmark that is aimed at assessing the ability of current VLMs to resolve TPRs—an important capability which is essential for establishing common ground.

## 6.3 The BlockWorld-Repairs Dataset

We use the corpus collected in Chapter 3 to build the BlockWorld-Repairs dataset, also based on the Blocks World environment (Section 3.3). As discussed in that chapter, dialogues include up to two corrections following the user instructions for each sub-task (i.e., source block and target position prediction), with a maximum of seven user turns, including confirmations.

The automated rule-based agent interacting with AMTurk workers occasionally struggled to interpret directions accurately from the first user correction, leading to unexpected actions for users, e.g., [User Instruction] $\rightarrow$ *"This block?"* $\rightarrow$ *"No, down and left, but above the first row of blocks"* $\rightarrow$ *"This one?"* $\rightarrow$ *"No, I said down and left, closer to before"*). In such cases, language nuances or misinterpretations caused breakdowns or diverged from cooperative exchange principles, potentially rendering any secondary corrections ineffective. Therefore, to assess model performance on handling and exploiting user repairs, we simplify these dialogues by dropping the second corrections and focus our scope to a single correction. Below, we describe the modifications made to the corpus to create the final dataset for our benchmark, followed by validating it through a human study.

### 6.3.1 Data

In addition to removing secondary corrections from our dialogues, we streamline the dataset structure for our experiments into a format aligned with other works in VLMs.

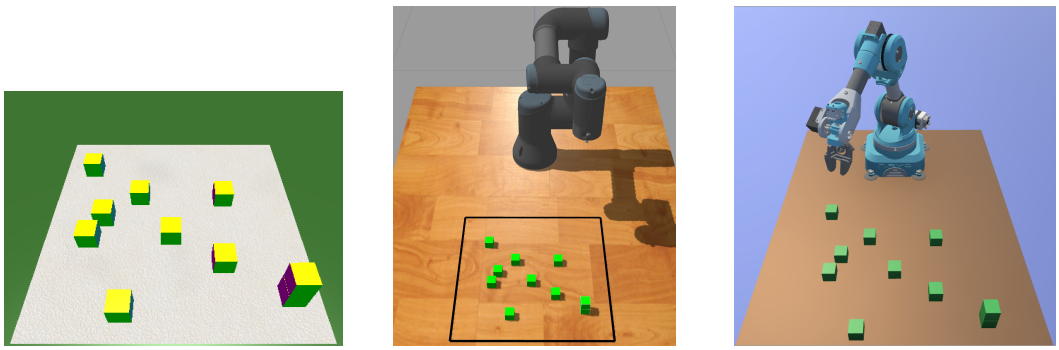
**Visual Features** Since the release of the Blocks World and our corpus collections, VLMs have standardised around using bounding boxes (a set of four pixel coordinates: $[x_0, y_0, x_1, y_1]$) to denote image regions. Object-detection tasks from computer vision (Lin et al., 2014a; Yu et al., 2016a) and subsequent vision transformers pre-trained on visual grounding objectives (Wang et al., 2022; Liu et al., 2024b; Chen et al., 2023) have adopted bounding boxes to process visual input due to the availability of large datasets and compatibility with model architectures. In contrast, the Blocks World dataset provides 3D coordinates, which few models are trained to handle (Hong et al., 2023) due to their complexity, making it challenging for standard VLMs to process our data out-of-the-box.

The simulation platform used in our corpus collection Section 3.4.1 focuses on robotics and real-world physics, enabling digital twin simulations and transfer to real-world settings. However, since then, more flexible simulation environments have emerged or matured. Some platforms, such as Habitat (Savva et al., 2019) or AI2Thor (Kolve et al., 2017), provide realistic environments for researching embodied AI, whereas others, like Unity[3], are more common for diverse applications. Unity, in particular, offers computer vision tools[4] for programmatically extracting visual features and collision metadata, which enables to generate object relational information (e.g., left of, on top of) and, crucially, bounding boxes.

Therefore, we simulate the Blocks World in Unity with PDSim (De Pellegrin and Petrick, 2024) to automatically generate block bounding boxes for training and evaluation, aligning with common visual grounding objectives in VLMs. We replicate the same block arrangements as in the original dataset and our collected corpus from the 3D coordinates. For this chapter's experiments, we will use images from Unity, which are much higher resolution than previous images (see Figure 6.2 for a comparison).

[3]Game development platform, see `https://unity.com/`

[4]See `https://unity.com/products/computer-vision`



(a) Bisk et al. (2016) (b) Chapter 3 (c) Unity (this chapter)

Figure 6.2: Comparison of the same block configuration across simulations.

**Dialogue Context** Instead of using full dialogues, we segment the corpus dialogues after each user turn to obtain more entries for training. We also remove the acknowledgement turns, which were obvious since the rule-based agent would always pick the correct position after the second correction, regardless of the utterance[5]. Additionally, we limit dialogues to one correction at a time, splitting dialogues that have corrections for both sub-tasks into two separate entries. Between corrections, we keep the agent's utterances but append bounding box coordinates for the block or position pointed by the robot (e.g., *"This one?* $[x_0, y_0, x_1, y_1]$*"*). In the end, we obtain three types of data entries:

(1) **Single-turn instructions:** 796 entries from all dialogue initial instructions (e.g., *"Move the block..."*). These entries are conceptually similar to the original Blocks World instructions (Bisk et al., 2016). Their prediction goal is both the true source block and target positions.

(2) **Source block corrections:** 629 entries from dialogues with at least one source block correction, containing the initial instruction, the agent's candidate response and the user correction (e.g., *"Move the block..."* $\rightarrow$ *"This block?* $[x_0, y_0, x_1, y_1]$*"* $\rightarrow$ *"No, the one directly above"*). Their prediction goal is only the source block bounding box.

(3) **Target position corrections:** 635 entries from dialogues with at least one correction for the target position, containing the initial instruction, the agent's candidate response and the user correction (e.g., *"Move the block..."* $\rightarrow$ *"Is here okay?*

[5]We manually checked dialogues for coherence, excluding those with nonsensical utterances from the corpus. However, we could not check every correction for directions towards the correct position, e.g., 'left' instead of 'right', and relied on AMTurk workers performing the task in good faith. We conduct a human study to validate instructions and corrections in Section 6.3.2 for this purpose.

$[x_0, y_0, x_1, y_1]$" → *"No, move slightly down and left"*). Their prediction goal is only the target position as a bounding box.

Altogether, this results in 2059 entries: 796 single-turn instructions and 1264 corrections, collectively referred to as Third Position Repair (TPR). Henceforth, we call this dataset the BlockWorld-Repairs or BW-R for short.

### *6.3.2 Human Study*

Since we used an automated agent to collect the dialogues and AMTurk data quality may vary (Saravanos et al., 2021), we conducted an in-person study to validate the data. This study aims to examine how humans would interpret and act on these instructions and repairs from the agent's perspective, establishing a baseline for human performance.

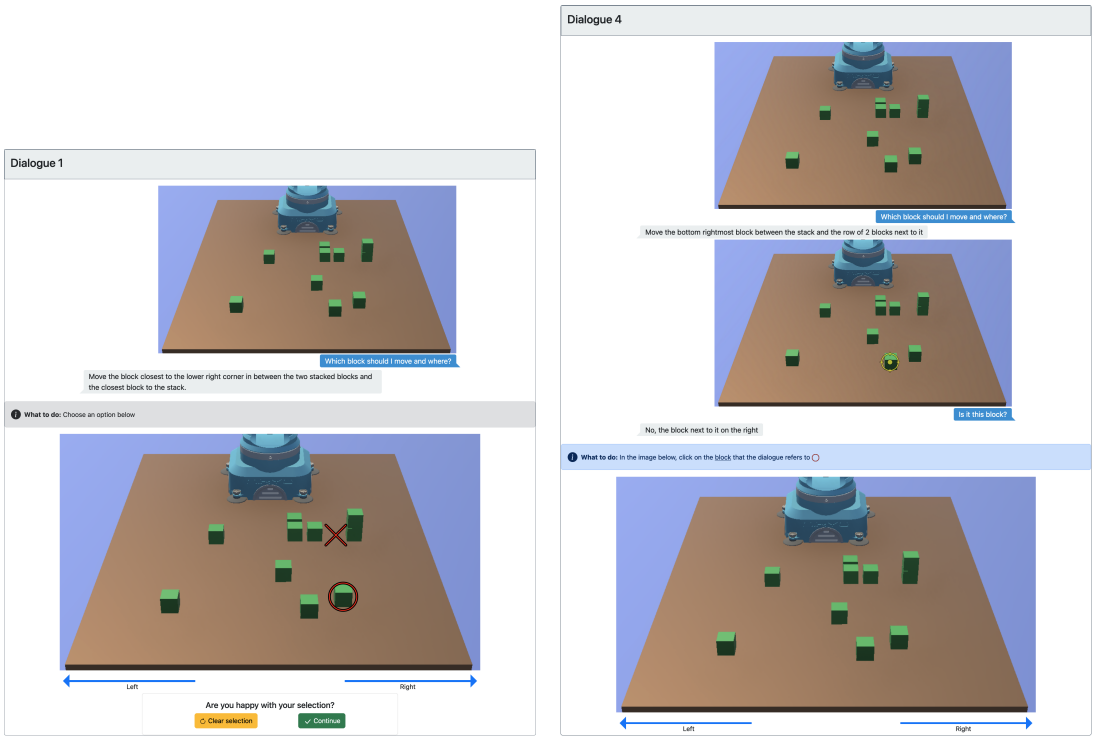
**Study Design** In Chapter 3, we elicited miscommunications in the Blocks World using a simulated environment, a rule-based agent and AMTurk users. To ensure a fair comparison between models and humans, we need to verify that humans can follow these instructions and repairs when placed in the opposite role. Thus, the goals of this study are to: 1) validate that the dialogues are coherent and interpretable; and 2) assess human performance within the same collaborative scenarios. We conducted a between-subjects study where participants reviewed dialogues from the BW-R dataset and selected a block to move and a position to place it based on the provided instructions.

**Task** Participants encountered two types of entries: 1) single-turn instruction dialogues, consisting of an initial message followed by an instruction identifying the source block and target position; and 2) dialogues with TPRs, which include the initial message and instruction, a candidate response and a correction. For single-turn instructions, participants had to select both the source block and target position, whereas for dialogues with TPRs, they selected only one of these (either source or target) depending on the correction.

Participants interacted with an interface similar to the data collection, with a dialogue entry displayed in a chat window, see Figure 6.3. They had to read the dialogue, and then click on a large image wherever they believed the dialogue referenced a block or a position within the table. Participants had no access to the true block configurations and received no feedback on whether they selected the correct block/location. Following our

data collection, we visually indicated the candidate block or position by overlaying lights on the images when a dialogue included a TPR.

Participants saw each dialogue once (e.g., if they saw the instruction-only segment, they would not see the corresponding TPR version), and we collected two annotations per dialogue entry. Dialogue entries are selected with a weighted randomiser algorithm, so that dialogues with existing annotations (from previous participants) have priority until they have two annotations each from different participants.




(a) Dialogue with a single instruction. (b) Dialogue with TPR.

Figure 6.3: User interface for the study.

**Measurements** We recorded the image pixels for the positions selected by participants, which we can convert to XYZ coordinates to calculate source block accuracy and block distances (see Section 3.3.1 for additional details on these metrics). We also measured the time taken to complete a dialogue as an indicator of task difficulty.

**Procedure** We conduct the study in a quiet university lab free from distractions (e.g., other people). Each participant was seated at a desk equipped with a 27-inch 4K monitor, keyboard and mouse. The study interface ran within a web browser.

We began by explaining the task to participants, who signed a consent form before

starting the study. Participants saw the task instructions along with a few example images. They proceeded to a training session with simpler dialogues to familiarise themselves with the controls (e.g., clicking on the image, clearing their selection or continuing to the next dialogue). Once they were ready, they would continue to the annotation task.

After the time had run out, the interface would stop showing dialogues, and participants continued to a final screen to complete a post-task questionnaire with demographic questions and space for free-form feedback. Each session would last a maximum of 40 minutes, including the post-task questionnaire, with the number of dialogue annotations completed limited only by time.

**Participants** We recruited 22 participants (5 women, 17 men, mostly in the age range 23-29) through institution mailing lists. Participants received £10 for taking part in the study. Our institution's ethics committee provided ethics approval for this study. The group included a mix of native and non-native English speakers, with 12 reporting English as their first language at native proficiency, and 10 indicating advanced or higher English proficiency. 13 participants were enrolled or had completed a Masters degree, with the remaining 9 enrolled in an undergraduate level degree. Recruiting from university settings likely yielded participants with above-average education and technology familiarity compared to the general population, potentially limiting generalisability. However, the observed variance in participant performance suggests that our sample still captured a meaningful range of annotation behaviors, supporting the broader applicability of our findings. We also do not train models on this dataset, avoiding any potential bias in the model-human comparisons.

**Results** After filtering out incorrect or problematic entries, we obtained 991 action annotations. Overall, participants performed well and rated the AMTurk dialogues highly. They achieved a mean accuracy of 68% in identifying the correct block to move, which improved to 75% after repairs (mean block distance also reduced from 1.59 to 1.41), with similar success in target position identification. Notably, performance variability decreased after repairs (standard deviation reduced from 14.8 to 12.2), highlighting that repairs improved performance across participants. These findings confirm that even in ambiguous scenarios, humans leverage repairs to solve tasks more effectively, emphasising the importance of models accurately processing TPRs. We will analyse human

performance and compare it with VLMs in Section 6.5.

### 6.3.3 Comparison to Related Situated Collaborative Datasets

We compare related datasets with the BlockWorld-Repairs in Table 6.2. Our benchmark is a robotic manipulation task involving common pick&place actions and rich referring expressions in a situated dialogue. Similarly, both TEACh (Padmakumar et al., 2022) and DialFRED (Gao et al., 2022) extend the popular ALFRED (Shridhar et al., 2020) to include dialogues with clarificational exchanges. From a high-level ALFRED directive (e.g., "prepare coffee"), these works crowd-source possible questions for low-level actions that may be underspecified in the original task (e.g., "where is the coffee mug?"). Thus, they often operate on the last level of Clark's (1996) joint action ladder, *action and consideration*, where the goal is already clear and lack of knowledge (underspecified location of objects or actions available) represents the primary ambiguity (e.g., VIMA-Bench; Jiang et al., 2023b)). These works rarely feature bi-directional information flow (*human* $\leftrightarrows$ *agent*) in their dialogues, unlike BW-R which ensures dialogue dependency with partial robot movements and candidate responses. DialFRED (Gao et al., 2022), Alexa Arena (Gao et al., 2023b) and CoDraw iCRs (Madureira and Schlangen, 2023) contain questions for underspecified actions/objects (e.g., "Which spoon should I pick up?") but, in these situations, the ambiguity is reduced to a minimum so that candidates are easily distinguishable with strong visual representations (e.g., "yellow or blue spoon?" or "left or right?").

By contrast, SIMMC 2.0 (Kottur et al., 2021)[6] includes a large number of visually ambiguous objects (e.g., 5 identical red t-shirts in view) within long dialogues that reference multiple items (averaging 4.5±2.4 unique objects per dialogue). This makes it rich in cross-modal coordination phenomena, with terse referring expressions that mix spatial, historical and visual cues (e.g., "Pick the red shirt. Which one? The one on the right wardrobe, above the blue jumper"). Similarly, Cups (Dobnik et al., 2020) aligns with our work, as it also explores coordination in highly ambiguous environments where many objects share the same shape/colour but are in different positions, requiring complex referring expressions. However, Cups focuses on Frame of Reference coordination rather than manipulation or resolving coreferences. BW-R is aimed at exploring this

[6]We previously discussed and used this dataset in Chapter 4 and Chapter 5.

coordination as part of a collaborative human-robot tabletop manipulation task.

| Related Work | Manipulation | Ref Exp | Dialogue | Repairs | Miscommunication |
|---|---|---|---|---|---|
| VIMA-Bench (Jiang et al., 2023b) | ✓ | R | ✗ | ✗ | - |
| ALFRED (Shridhar et al., 2020) | ✓ | R | ✗ | ✗ | - |
| TEACh (Padmakumar et al., 2022) | ✓ | V R | ✓ | ✓ | (4) Action |
| DialFRED (Gao et al., 2022) | ✓ | V R | ✓ | ✓ | (4) Action |
| Alexa Arena (Gao et al., 2023b) | ✓ | V R | ✓ | ✗ | (4) Action |
| SIMMC 2.0 (Kottur et al., 2021) | ✗ | V R D | ✓ | ✓ | (3) Understanding |
| CoDraw iCR (Madureira and Schlangen, 2023) | ✗ | V R | ✓ | ✓ | (4) Action |
| Cups (Dobnik et al., 2020) | ✗ | V R D | ✓ | ✓ | (3) Understanding |
| Block World (Bisk et al., 2016) | ✓ | V R | ✗ | ✗ | - |
| **BLOCKWORLD-REPAIRS (Ours)** | ✓ | V R | ✓ | ✓ | **(3) Understanding** |

Table 6.2: Comparison of related works to our dataset BlockWorld-Repairs. Columns describe whether the datasets have: (1 **Manipulation**) robotic manipulation as a task (i.e., pick&place); (2 **Ref Exp**) referring expression type following Section 5.4, R for relational expressions (e.g., "the left one"), V for references to visual properties (e.g., "the green one") or D when referring to the dialogue history (e.g., "the one I mentioned"); (3 **Dialogue**) dialogue with multiple turns for the same action; (4 **Repairs**) explicit repairs in the data (i.e., clarifications or corrections); and (5 **Miscommunication**) the miscommunication level based on Clark's (1996) joint action ladder.

## 6.4 Experiments on Learning to Process Repairs

This section assesses VLMs' ability to process instructions and subsequent TPRs within a table-top manipulation task in the Blocks World. In BW-R, the repair turns are only meaningful when interpreted alongside the robot's candidate position at the time of the correction, based on the initial instruction. Therefore, are incomplete on their own. Consequently, we have dialogue triplets intrinsically linked and can only be comprehended as a whole: the initial instruction, the incorrect candidate prediction, and the repair. We argue that more general models must be able to effectively process both initial instructions and any subsequent repairs.

### 6.4.1 Experimental Setup

We provide the most relevant details of the experimental setup in this section but refer to Appendix B for additional information.

#### Dataset

We combine the Blocks World single-turn instructions (Bisk et al., 2016; Bisk et al., 2018) with our BW-R data, using a 70/30 train/test split. This yields a total of 3613 samples for training and 1659 for testing. We maintain the block configurations from the original Blocks World and evaluate models on unseen world arrangements. In our experiments, we ablate the data to gain fine-grained insights into the models' performance and capabilities.

We do not set aside a validation set for these experiments as we do not perform hyper-parameter tuning of the VLMs. Following standard practice with parameter-efficient fine-tuning of LLMs (e.g., LoRA (Hu et al., 2021)), we evaluate models only at two points: before training (zero-shot) and after training completion (fine-tuned). This approach prevents overfitting to the test set while keeping the VLMs' hyperparameters consistent, minimising the data required for the experiments.

#### Task Formalisation

We evaluate models on the Blocks World sub-tasks (Section 3.3.1): 1) **source block prediction**, identifying the block to pick up; and 2) **target position prediction**, identifying where to place the block. Prior works have formulated these tasks in a variety of ways to aid predictions, such as locating a reference block and predicting an offset distance from it (Bisk et al., 2016), restricting predictions to board quadrants (Mehta and Goldwasser, 2019) or using adversarial data augmentation (Dan et al., 2021). In this section, we instead formulate the task as bounding box prediction within the image for both the block and its final location to align with VLMs' visual grounding pre-training.

We prompt models to generate the bounding box coordinates ($[x_0, y_0, x_1, y_1]$) normalised to the 0-1 range, which we then scale to the image dimensions (1024x576 pixels). Figure 6.4 shows the pipeline for the task, including how we derive the metrics to assess the models' performance. Predictions not matching the expected format during parsing are marked as out-of-distribution and excluded. In such cases, we use the output of a

random baseline to calculate the metrics for that particular entry[7].

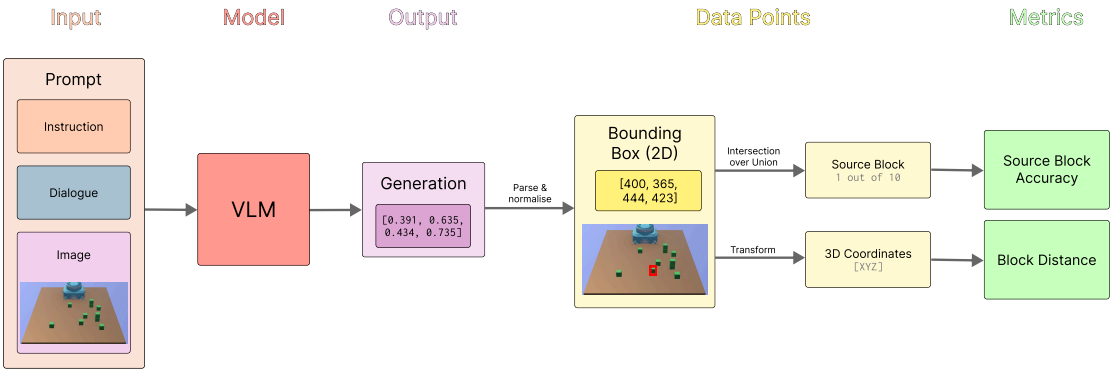



Figure 6.4: Task pipeline with the VLMs. We transform the model outputs into their equivalent 3D coordinates in the virtual world, then calculate metrics from the Blocks World (Section 3.3.1) such as block distance. Sample prompts and outputs are provided in Appendix B.

### Metrics

For evaluation, we use metrics originally introduced in the Blocks World (Bisk et al., 2016). For **source block accuracy**, we use Intersection over Union and select blocks above a 0.5 threshold, following standard practice in object segmentation tasks. To measure **normalised block distances**, we convert bounding boxes to their respective XYZ coordinates in the virtual world, then calculate the distance between predicted and true locations. Converting 2D bounding boxes to 3D coordinates introduces a slight alignment error due to the image projection angle and virtual world grid. In our initial calibration tests, we found a discrepancy of 0.1 block distance between the centre of a predicted block and its true XYZ coordinates in the virtual world, which is negligible for our task results. Refer to Section 3.3.1 for further metric details. When models predict invalid bounding boxes, we instead select a block or XYZ position from a random baseline.

### Models

We fine-tune and experiment with two state-of-the-art open-source models, Idefics2 (Laurençon et al., 2024) and LLaVA 1.5 (Liu et al., 2024a), and additionally evaluate GPT-4o (OpenAI, 2024), a frontier AI[8] VLM, in a zero-shot setting.

[7] These entries constituted approximately less than 3% of the test predictions depending on the model.

[8] See https://www.gov.uk/government/publications/frontier-ai-capabilities-and-risks-discussion-paper [Last accessed: 6 Nov 2024]

**Idefics2 7B** A general-purpose VLM (Laurençon et al., 2024) that combines a Mistral 7B (Jiang et al., 2023a) language backbone with a SigLIP-SO400M visual encoder (Zhai et al., 2023). It first encodes images through the vision encoder, then concatenates the resulting output with the input text tokens for a fully autoregressive architecture without cross-modal attention (see Figure 6.5). The model is pre-trained on a large mix of text and image-text data, followed by fine-tuning with a multi-modal instruction dataset, The Cauldron (Laurençon et al., 2024). After unsatisfactory preliminary tests with the "chatty" version of Idefics2 that was trained on dialogues, we decided to use the instruction-tuned version instead.

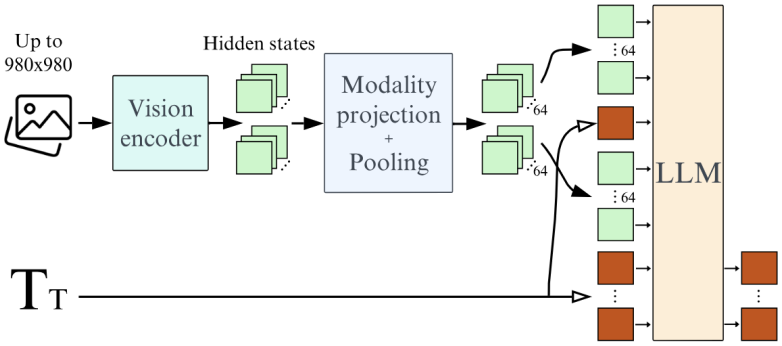



Figure 6.5: Idefics2 architecture from Laurençon et al. (2024).

**LLaVA 1.5 7B** A strong general-purpose VLM (Liu et al., 2024b) that integrates a Vicuna (Chiang et al., 2023) language decoder (based on LLaMA (Touvron et al., 2023a)) with a pre-trained CLIP visual encoder ViT-L/14 (Radford et al., 2021). Similar to Idefics2, LLaVA encodes images and concatenates the visual encoder output with text tokens for autoregressive token prediction (see architecture in Figure 6.6). We use version 1.5 of LLaVA (Liu et al., 2024a), which replaces the linear projection of the visual output with a multi-layer perceptron (MLP) (Chen et al., 2020a) to improve the quality of the learned representations. The model is trained on data generated from GPT-4 with a particular focus on image descriptions, reasoning and conversational tasks.

**Task Training** In our experiments, we use the VLMs' default hyperparameters and fine-tune with their recommended parameter-efficient methods: LoRA (Hu et al., 2021) for LLaVA and QLoRA (Dettmers et al., 2023) for Idefics2. We apply each model's default image processing pipeline with our images of size 1024 by 576 pixels. We train for 1 epoch, as additional epochs showed no benefit, with batches containing a random

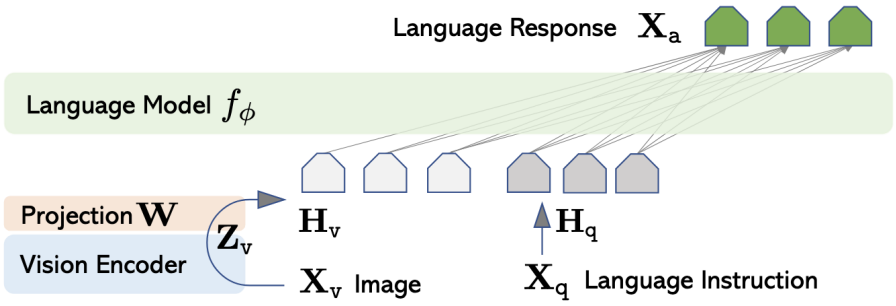



Figure 6.6: LLaVA architecture from Liu et al. (2024b).

mix of instructions and repairs. To avoid interference between objectives (i.e., identifying an object for source block prediction or an empty space for target position prediction) and fully leverage the input prompt, we train separate models for each sub-task, leaving multi-task learning for future work. The input prompts include task instructions, the dialogue, and the relevant board image, as outlined in Appendix B.

### *6.4.2 Training Approaches*

Repairs are a complex dialogue phenomenon: they are often incomplete or terse utterances that follow Grice's maxim of Quantity (Grice, 1975) ("give as much information as needed, and not more") and can only be understood within the current interaction context. The information needed to correctly ground referring expressions may be distributed across multiple dialogue turns or require context, such as a previous candidate object. Thus, models cannot solely rely on the most recent dialogue turn; they must consider the initial user instruction and any following candidate output, like a proposed bounding box, as the repair may be relative to this candidate's position (e.g., *"No, the one to the right"*).

Typically, supervised LLM fine-tuning involves calculating the cross-entropy loss for each token in the input prompt, so models' generation aligns more closely to the actual tokens (Laurençon et al., 2024). Following this standard fine-tuning approach leads the models to learn from text sequences that include intermediate, incorrect bounding boxes (candidate outputs as language tokens), alongside a final, correct bounding box. We hypothesise that learning from these incorrect bounding box tokens impairs the model's ability to master the intended task, negatively impacting generation quality and ultimately limiting their generalisation capabilities. Thus, we explore alternative training regimes to mitigate this issue and ultimately learn to jointly process instructions and repairs.

### Masking Token Loss

Inspired by concurrent works that explore custom masking criteria (Huerta-Enochian and Ko, 2024), we design the following masking approaches to mitigate cross-entropy loss on incorrect candidate bounding boxes (also see Figure 6.7):

- **Default Loss:** standard LLM cross-entropy loss applied to each input token (Laurençon et al., 2024).
- **User-Turn Loss:** all utterances that do not originate from the user are masked, preventing models from learning from incorrect tokens, such as candidate bounding box predictions. This approach ensures that models benefit from learning the task or dialogue history, whilst avoiding calculating cross-entropy losses on the inaccurate intermediate outputs.
- **Completion-Only Loss:** involves masking all tokens except for the last expected output response. Consequently, models do not learn from previous dialogue turns and direct their outputs specifically towards the correct bounding boxes. We anticipate that these models will show strong generalisation capabilities.

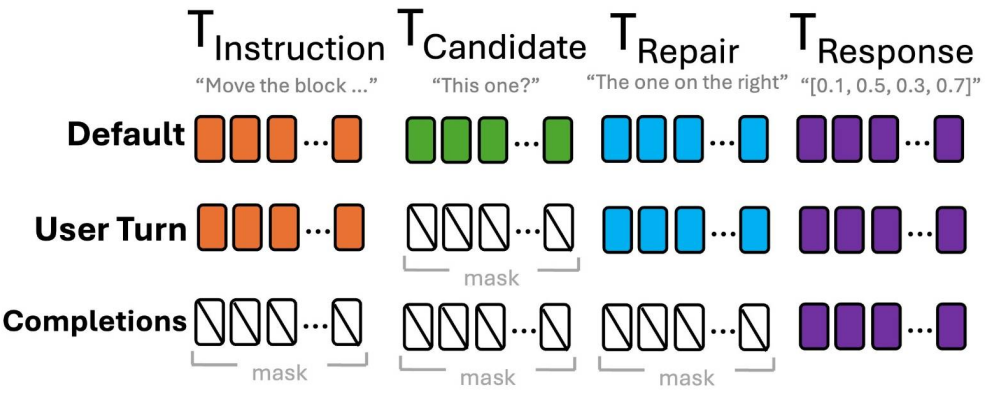



Figure 6.7: Custom masking criteria to mitigate cross-entropy loss on incorrect tokens.

### 6.4.3 Results

TPRs are fundamental elements of human dialogues, so we expect VLMs to effectively process them along with single-turn instructions. To assess their capabilities, we conduct experiments with VLMs across three distinct loss training regimes (default, user-turn and completions-only) and data ablations, including zero-shot evaluations for all models and GPT-4o. We fine-tune two models (Idefics2 and LLaVA), separately for each sub-task for 1 epoch: source block prediction and target position prediction. We present all results in Table 6.3 and discuss them in detail below.

**Zero-shot** VLMs exhibit some out-of-the-box capabilities to handle instructions and repairs in the Blocks World. They typically perform above the random baseline, demonstrating their grounding capabilities. GPT-4o is unique in its ability to effectively process instructions and subsequent repairs.

**Default Fine-tuning** We are able to distil the capability to process instructions and repairs via standard cross-entropy losses, where each prompt token is part of the loss calculation. Fine-tuned models generally learn the task and perform well when evaluated on the same data that they were trained (instructions or repairs). However, fine-tuning with a combination of both entries does not consistently improve performance: Idefics2 seems to adapt well to the training data and shows greater data efficiency when training and testing on data subsets, whereas LLaVA achieves better results with the full data and therefore suggests stronger generalisation capabilities.

Overall, these models struggle to exploit TPRs in these dialogues. Intuitively, repairs should facilitate the task by introducing new information that refines the candidate pool. However, because repairs are often context-dependent (i.e., based on the candidate response), models require strong object-centric representations (Bengio et al., 2013; Seitzer et al., 2023) and a deeper understanding of relational dynamics between objects to effectively leverage TPRs (e.g., "the block to the right") (Ilinykh and Dobnik, 2021b). This is a particular issue of current vision transformers, which frequently struggle with positional concepts such as spatial understanding (Ilinykh and Dobnik, 2022; Pantazopoulos et al., 2024), and tend to rely on language biases (Salin et al., 2022).

| Test Data | Train Data | Model | Loss | Source Accuracy ↑ | Source Distance (SD) ↓ | Target Distance (SD) ↓ |
|---|---|---|---|---|---|---|
| Single-Turn Instructions | | Idefics2 *zs* | Default | 0.18 | 5.06 (±3.50) | 9.55 (±2.50) |
| | Instructions | Idefics2 | Default | 0.33 | 3.33 (±3.02) | 4.93 (±2.64) |
| | Instructions | Idefics2 | User-turn | 0.33 | 3.33 (±3.02) | 4.93 (±2.64) |
| | Instructions | Idefics2 | Completions | 0.32 | 3.44 (±3.04) | 4.75 (±2.62) |
| | Repairs | Idefics2 | Default | 0.21 | 4.36 (±3.13) | 5.38 (±2.44) |
| | Repairs | Idefics2 | User-turn | 0.22 | 4.59 (±3.25) | 8.03 (±3.21) |
| | Repairs | Idefics2 | Completions | 0.14 | 4.88 (±3.07) | 6.19 (±2.97) |
| | Full | Idefics2 | Default | 0.30 | 3.39 (±2.92) | 4.75 (±2.60) |
| | Full | Idefics2 | User-turn | 0.36 | 3.12 (±3.00) | 4.59 (±2.59) |
| | Full | Idefics2 | Completions | 0.33 | 3.33 (±2.99) | 4.52 (±2.58) |
| | | LLaVA *zs* | Default | 0.24 | 4.40 (±3.34) | 5.74 (±3.08) |
| | Instructions | LLaVA | Default | 0.25 | 3.77 (±2.99) | 5.06 (±2.92) |
| | Instructions | LLaVA | User-turn | 0.27 | 3.79 (±3.03) | 5.02 (±2.86) |
| | Instructions | LLaVA | Completions | 0.27 | 3.76 (±3.00) | 4.40 (±2.69) |
| | Repairs | LLaVA | Default | 0.14 | 4.97 (±3.24) | 6.24 (±2.91) |
| | Repairs | LLaVA | User-turn | 0.18 | 4.19 (±2.82) | 6.37 (±2.91) |
| | Repairs | LLaVA | Completions | 0.09 | 5.12 (±3.00) | 6.27 (±2.86) |
| | Full | LLaVA | Default | 0.22 | 3.82 (±2.91) | 4.46 (±2.49) |
| | Full | LLaVA | User-turn | 0.24 | 3.65 (±2.86) | 4.60 (±2.54) |
| | Full | LLaVA | Completions | 0.19 | 3.93 (±2.73) | 4.51 (±2.82) |
| | | GPT-4o *zs* | | 0.30 | 3.83 (±3.38) | 4.22 (±2.65) |
| Repairs | | Idefics2 *zs* | Default | 0.00 | 3.74 (±1.95) | 4.13 (±2.05) |
| | Instructions | Idefics2 | Default | 0.21 | 4.19 (±2.75) | 5.11 (±2.60) |
| | Instructions | Idefics2 | User-turn | 0.26 | 3.74 (±2.84) | 4.68 (±2.12) |
| | Instructions | Idefics2 | Completions | 0.28 | 3.66 (±2.77) | 3.98 (±2.58) |
| | Repairs | Idefics2 | Default | 0.58 | 2.19 (±3.06) | 7.07 (±2.77) |
| | Repairs | Idefics2 | User-turn | 0.58 | 2.16 (±2.92) | 6.45 (±2.22) |
| | Repairs | Idefics2 | Completions | 0.57 | 1.55 (±1.95) | 7.35 (±2.02) |
| | Full | Idefics2 | Default | 0.26 | 3.43 (±2.40) | 5.81 (±2.35) |
| | Full | Idefics2 | User-turn | 0.32 | 2.94 (±2.56) | 6.00 (±1.95) |
| | Full | Idefics2 | Completions | 0.47 | 2.29 (±2.36) | 5.90 (±2.71) |
| | | LLaVA *zs* | Default | 0.13 | 3.29 (±2.09) | 3.45 (±1.20) |
| | Instructions | LLaVA | Default | 0.18 | 3.43 (±2.49) | 3.49 (±1.34) |
| | Instructions | LLaVA | User-turn | 0.16 | 3.47 (±2.42) | 3.46 (±1.39) |
| | Instructions | LLaVA | Completions | 0.25 | 2.79 (±2.47) | 3.76 (±1.88) |
| | Repairs | LLaVA | Default | 0.07 | 3.20 (±1.79) | 3.67 (±1.31) |
| | Repairs | LLaVA | User-turn | 0.13 | 3.28 (±2.05) | 3.89 (±1.35) |
| | Repairs | LLaVA | Completions | 0.50 | 2.33 (±2.64) | 5.57 (±1.67) |
| | Full | LLaVA | Default | 0.44 | 2.66 (±2.76) | 3.69 (±2.18) |
| | Full | LLaVA | User-turn | 0.37 | 2.69 (±2.48) | 4.04 (±2.41) |
| | Full | LLaVA | Completions | 0.54 | 2.01 (±2.47) | 4.56 (±2.69) |
| | | GPT-4o *zs* | Default | 0.41 | 2.75 (±3.06) | 2.95 (±2.17) |
| | | Random Baseline | | 0.10 | 6.50 (±3.0) | 6.26 (±2.9) |

Table 6.3: Results of the VLMs' performance on source and target sub-tasks across training regimes and zero-shot (zs) using the BW-R benchmark. We ablate training and test data into instructions-only, repairs-only, or the full data. Metrics include **source block accuracy** (↑) and **mean block distance** (↓), where lower distances indicate predictions closer to the correct location.

**Masking Losses** We first observe that alternative loss-masking approaches have minimal impact when evaluating single-turn instructions, although both VLMs show slight improvements when trained on the full dataset with user-turn losses, suggesting potential benefits for generalisation. Notably, both models exhibit substantial gains in source block prediction when tested on repairs, particularly when fine-tuning with only repairs or the full dataset.

These alternative loss approaches enable models to learn from TPRs and transfer some task knowledge from instructions to repairs when using the full dataset. Calculating losses on completions show significant benefits for repairs, but may slightly hinder VLMs performance when trained and tested on the same data. This is expected, as calculating the loss across the full input allows models to leverage more tokens during training, which facilitates learning the input format and quickly adapting to the task. However, default loss calculations also seems to hinder deeper task learning, limiting the VLMs' generalisation capacity for instructions and repairs.

We assumed that the intermediate candidate responses hinder model performance, as they contribute incorrect information to the default cross-entropy loss calculation. When bounding boxes appear in the input text, models attend to and learn to predict them, steering the predictions towards incorrect locations that have been repaired. Masking these tokens (user-turn loss) provides slight benefits when evaluating instructions alone, but produces mixed overall effects (positive or negative). In contrast, completions-only loss shows substantial improvements for handling repairs, with only a minor reduction in performance on instructions.

We also find that most improvements from either loss-masking approach primarily impact the source block prediction sub-task. VLMs struggle with target position predictions and do not show clear benefits from these losses. While both sub-tasks expect a similar bounding box output, predicting a target location is inherently more challenging, as it relies on identifying *spaces* between blocks rather than a salient object (e.g., a block). It is thus unsurprising that VLMs pre-trained with visual grounding objectives (e.g., RefCOCO (Yu et al., 2016a)) do not perform well when the task involves spatial locations rather than specific objects. In these cases, additional prompt tokens appear more beneficial to learning the task, and masking them can degrade performance.

In these fine-tuning experiments, we identify two key factors affecting performance: 1) the size of the training data; and 2) the incorrect candidate responses. When training

on repairs, the smaller data size (1) becomes the bigger issue as masking considerably reduces the available tokens during training, which is especially critical with limited data. Models do better when there is more training data, as happens with instructions, but they train on incorrect bounding boxes from the intermediate system turn (2), hurting their performance. Using completions-only losses helps models develop a more robust representation, improving their ability to generalise across both types of data.

## 6.5 Error Analysis

Our goal is to develop models capable of collaborating effectively with humans towards a shared objective, even when misunderstandings occur. To achieve this, we need a deeper understanding of how both models and humans approach the same visual-grounding task. In this section, we analyse models and humans beyond performance metrics, identifying mismatches and exploring how these differ from human processing of TPRs. For our comparisons, we focus on the VLMs with the strongest generalisation capabilities from Table 6.3, specifically those fine-tuned with completions-only losses on the full dataset. Furthermore, to ensure a fair comparison, we evaluate both models and humans on the same data subset (BW-R entries with human annotations), which may produce results slightly different from those in Table 6.3

### *6.5.1 Human-Model Comparison*

We begin by comparing the performance of models before and after fine-tuning to humans in Table 6.4. Clear differences emerge between VLMs abilities, which generally find this task challenging in zero-shot settings (except for GPT-4o). Both Idefics and LLaVA show substantial improvement with completions-only loss during fine-tuning, with the latter VLM reaching near-human performance levels in processing source block TPRs.

For single-turn instructions, improvements are more modest across models. In these cases, accuracy alone does not capture prediction quality, as lower mean block distances provide a more accurate reflection of true proximity. While results on single-turn instructions leave room for improvement, our focus is on a more generalised approach that effectively handles both instructions and their corrections, rather than aiming for perfect instruction understanding.

| Test Data | Model | Source | | Target |
|---|---|---|---|---|
| | | Accuracy ↑ | Block Distance (SD) ↓ | Block Distance (SD) ↓ |
| Instructions | Idefics2 - *zero-shot* | 0.18 | 6.31 (±4.4) | 9.84 (±2.0) |
| | Idefics2 - *fine-tuned* | 0.21 | 3.97 (±2.8) | 5.52 (±3.1) |
| | LLaVA - *zero-shot* | 0.21 | 6.08 (±5.5) | 5.06 (±5.4) |
| | LLaVA - *fine-tuned* | 0.16 | 4.08 (±4.1) | 4.63 (±3.4) |
| | GPT-4o | 0.26 | 2.96 (±3.7) | 4.30 (±3.4) |
| | Human Participants | 0.68 | 1.59 (±1.6) | 3.64 (±3.4) |
| Repairs | Idefics2 - *zero-shot* | 0.00 | 4.09 (±2.4) | 3.86 (±1.7) |
| | Idefics2 - *fine-tuned* | 0.43 | 2.46 (±2.3) | 5.80 (±2.7) |
| | LLaVA - *zero-shot* | 0.14 | 3.42 (±2.3) | 3.45 (±1.1) |
| | LLaVA - *fine-tuned* | 0.60 | 1.82 (±2.5) | 4.82 (±2.7) |
| | GPT-4o | 0.50 | 2.42 (±3.2) | 2.72 (±1.8) |
| | Human Participants | 0.75 | 1.41 (±1.4) | 2.77 (±2.8) |

Table 6.4: Humans compared to VLMs for the BW-R test subset with human annotations.

The target position sub-task poses a greater challenge for both humans and models, reflected in worse scores overall. Fine-tuning negatively affects both LLaVA and Idefics2 models' performance on TPRs, which already struggle with repairs related to target positions. This contrasts with GPT-4o, which outperforms human participants in processing TPRs. Notably, while fine-tuning improves performance on instructions, we observe that it fails to help models with their predictions for repairs, suggesting these models may be capable of learning the sub-task of target position prediction but cannot effectively handle the repairs themselves. Regardless of training method (see Table 6.3), both models appear to incorrectly learn target location prediction, performing worse than in zero-shot evaluation on TPRs.

### *6.5.2 Task Difficulty Analysis*

To identify the strengths and weaknesses of each models, we categorise the BW-R data into difficulty levels according to human performance[9]. Lower human performance serves as an indirect indicator of example complexity, allowing us to define three difficulty levels:

(1) **Easy:** Both human annotators of the same entry correctly predicted the source

[9]We provide this analysis according to GPT-4o performance in Appendix B

block (100% accuracy), or their predicted target position was within 1 block unit from the true location.

(2) **Medium:** At least one out of the two annotators correctly predicted the source block (50% accuracy), or their predicted target position was on average below the human's mean distance from the true location (between 1 and 3.22 units).

(3) **Hard:** Neither of the annotators correctly predicted the source block (0% accuracy), or their predicted target position exceeded the human mean distance (more than 3.22 units away).

These subsets have distributions of 68/11/21% for source block, and 30/36/32% for target prediction sub-tasks for easy/medium/hard respectively. The thresholds to split target prediction by difficulty were selected to create three roughly equal subsets. However, source block prediction was considerably easier for humans, resulting in an overwhelming majority with perfect scores that prevented a balanced subset distribution.

| **Level** | **Model** | **Source** | | **Target** |
|---|---|---|---|---|
| | | **Acc ↑** | **Dist (SD) ↓** | **Dist (SD) ↓** |
| Easy | Idefics2 | 0.35 | 3.09 (±3.1) | 5.72 (±3.1) |
| | LLaVA | 0.41 | 2.62 (±2.6) | 3.98 (±2.5) |
| | GPT-4o | 0.45 | 3.50 (±3.5) | 2.18 (±1.3) |
| | Humans | 1.00 | .00 (±.0) | .81 (±.5) |
| Medium | Idefics2 | 0.37 | 2.32 (±2.3) | 5.30 (±2.3) |
| | LLaVA | 0.42 | 2.19 (±2.2) | 4.54 (±3.0) |
| | GPT-4o | 0.42 | 3.05 (±3.1) | 2.86 (±2.0) |
| | Humans | 0.50 | 2.24 (±2.2) | 2.28 (±1.3) |
| Hard | Idefics2 | 0.22 | 3.82 (±3.8) | 6.00 (±3.2) |
| | LLaVA | 0.25 | 4.30 (±4.3) | 5.77 (±3.3) |
| | GPT-4o | 0.16 | 4.46 (±4.5) | 5.30 (±3.7) |
| | Humans | 0.00 | 5.62 (±5.6) | 6.25 (±3.3) |

Table 6.5: Model and human performance across difficulty subsets (*easy, medium, hard*).

The evaluation of VLMs on these difficulty subsets (Table 6.5) reveals that models and humans encounter challenges with different entries. Models show a more uniform performance distribution, outperforming humans on the most difficult subset but significantly underperforming on easier examples. We also observe a trend related to the average word

count per subset entry (Table 6.6). Easier examples tend to have higher word counts, suggesting that these have longer, more descriptive phrases for references; whereas harder examples (with fewer words) may have easier terms for models, but are too underspecified for humans.

| Difficulty Level | Source Block | Target Position |
|---|---|---|
| Easy | 28.69 (±12.20) | 32.19 (±14.66) |
| Medium | 28.80 (±12.97) | 31.27 (±12.86) |
| Hard | 27.62 (±12.19) | 27.46 (±12.35) |

Table 6.6: Average word count and standard deviation (SD) for test dialogues in BW-R.

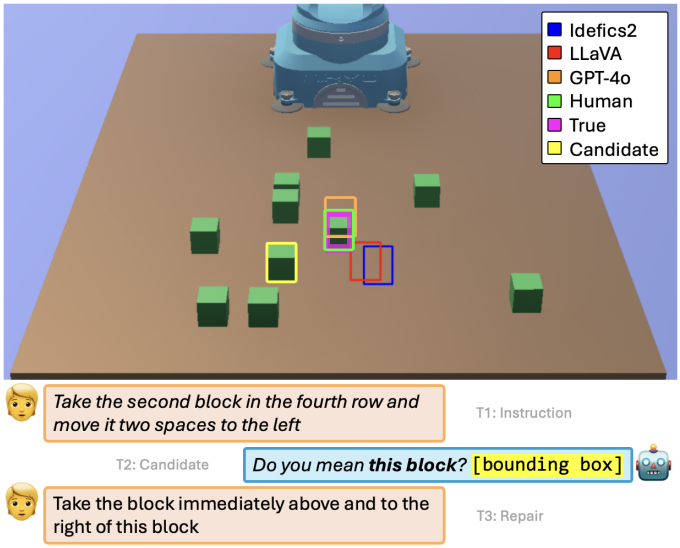
### *6.5.3 Qualitative Analysis*

To further examine proficiency differences between VLMs and humans, we manually analysed 50 BW-R dialogues with TPRs and provide examples in Figure 6.8. We observe that GPT-4o tends to make similar predictions to humans when processing candidate responses and repairs. While models can usually handle simple spatial repairs (e.g., left, right, above) (Chiyah-Garcia et al., 2023) and discussed in Chapter 5, they particularly struggle with more abstract concepts[10] (e.g., rows, columns) (Ilinykh et al., 2022).

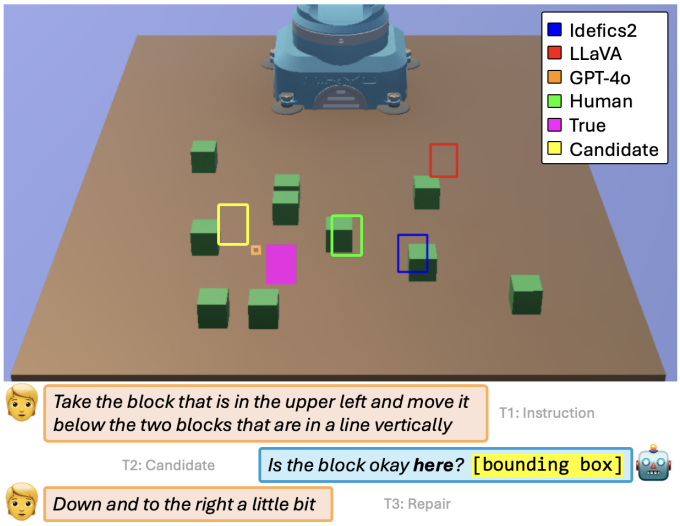
Both Idefics2 and LLaVA demonstrate some understanding of basic referring expressions, but struggle with longer or more complex sequences. Idefics2, in particular, seems prone to over-correct (e.g., interpreting *down* as moving to the bottom of the image), and both VLMs often predict bounding boxes not aligned with actual blocks in the source prediction sub-task, which is unexpected given their pre-training and fine-tuning with visual-grounding objectives. The presence of visually indistinguishable blocks may be confusing the models, similar to the challenges faced in Chapter 4 where models struggled to disambiguate similar-looking objects. These findings also highlight the limitations of source block accuracy as a metric (e.g., as in Table 6.3), making block distances a more reliable measure of prediction quality (closer to the true block).

In the target prediction sub-task, VLMs significantly struggle and largely fail to process repairs effectively. GPT-4o is the exception, which generally demonstrates a more fine-grained understanding of the task, though it underperforms on simpler dialogues

[10]We previously discussed in Section 3.4 how 17.5% of human utterances contained these in our corpus.




(a) Source block selection




(b) Target position prediction

Figure 6.8: Two medium-difficulty dialogues with the bounding boxes predicted by the VLMs and humans.

(Table 6.5).

In both tasks, we observe that GPT-4o is more aligned with human responses than the other models, not only achieving scores closer to humans, but making errors that are more similar to human errors. It is the only model that demonstrates above-average performance compared to other models on the notoriously difficult target prediction sub-task, despite a lack of training in a zero-shot setting.

## 6.6 Reinforcement Learning for Resolving Miscommunications

Our evaluations showed that models are far from human performance in the same situated dialogue task, despite the improvements from masking token losses. In this section, we briefly explore the potential of reinforcement learning techniques to further align models with human preferences and enhance their generalisability and downstream performance in resolving the BW-R challenging referring expressions.

Achieving coordination in a dialogue is not a trivial task. It requires interlocutors to assimilate what has been said so far, resolving anaphoric references, grounding objects or locations in the surrounding environment and, finally, grounding the dialogue itself by producing a relevant utterance. While humans are able to handle all of this effortlessly, our results show that VLMs struggle to fully understand the dialogues, resulting in suboptimal performance compared to our human baseline.

As previously discussed, GPT-4o shows a stronger alignment with human responses than the other models. We suspect this is due to its additional Reinforcement Learning from Human Feedback (RLHF) training (Stiennon et al., 2020; Gao et al., 2023a) using Proximal Policy Optimisation (Schulman et al., 2017). LLMs trained with RLHF (Ouyang et al., 2022; Ziegler et al., 2019) have shown remarkable success by training on human preferences, aligning model outputs with human judgments and further improving their capabilities in downstream tasks. In this section, we show with a case study that further fine-tuning a VLM with RLHF can align to the human's expected responses and improve its performance in the BW-R task.

### *6.6.1 Approach*

In Section 6.4.2 experiments, we examined the effects of masking irrelevant tokens from the input to avoid calculating losses from them. The default VLM sequence-to-sequence fine-tuning procedures involve calculating cross-entropy over every input token, so that the model learns to predict the task instructions and context given (including the dialogue), conditioning on the task itself. This aids models to adapt to downstream tasks faster and with fewer data than training on just the expected output. On the other hand, our masking helped models generalise to the overall task, finding a source block and a target location, by masking tokens related to the candidate responses. Since the candi-

dates had the same format as the completions (i.e., both bounding boxes like `[0.123, 0.456, 0.000, 0.123]`), the models would condition on these incorrect tokens, learning to predict wrong bounding boxes.

In supervised fine-tuning, sequence-to-sequence models predict tokens and then calculate a loss for each one of them, back-propagating the result as needed. However, some works have questioned the use of cross-entropy losses for learning visual grounding tasks with sequence-to-sequence VLMs (Carion et al., 2020). For instance, in our case, a model may predict a bounding box coordinate (`[0.291]`) differing only by a single digit from the expected answer (`[0.891]`), resulting in a small loss. However, these two coordinates are almost in opposites sides of the image despite their relatively small change. Thus, some tokens are more important than others, specifically the starting digits of each bounding box coordinate. This behaviour hinders how well VLMs can learn tasks with supervised fine-tuning.

Contrastive loss or learning (Shi et al., 2020; Udandarao et al., 2020) is a promising alternative to cross-entropy losses for training models. It consists on models comparing two or more proposed outputs and learning to choose the better one of the two. During training, models update their parameters towards the preferred output to align with particular human preferences. However, to effectively use contrastive loss, it requires collecting extensive human preference data, which is costly and time-consuming. Thus, here we explore using DPO (discussed in Section 2.2.2) (Rafailov et al., 2024) with BW-R data to further fine-tune models.

### *6.6.2 Experimental Setup*

In these experiments, we look into further fine-tuning Idefics2 (Laurençon et al., 2024) as a proof of concept for applying DPO to the BW-R benchmark. We use the same task and metrics as in Section 6.4.1.

#### Generating Contrastive Data

To generate contrastive examples for training with DPO, we first identify the gold prediction, referred to as *chosen* in DPO. This prediction corresponds to a bounding box, depending on the BW-R sub-task: either the source block to pick up or the target position to drop it off. The alternative *rejected* output is then generated as follows:

| Test Data | Model | Source Accuracy ↑ | Source Block Distance (SD) ↓ |
|---|---|---|---|
| Inst. | Idefics2 - *zero-shot* | 0.18 | 6.31 (±4.4) |
| | Idefics2 - *supervised fine-tuning* | 0.32 | 3.24 (±2.9) |
| | Idefics2 - *supervised fine-tuning + DPO* | **0.42** | **2.82 (±3.1)** |
| Repairs | Idefics2 - *zero-shot* | 0.00 | 4.09 (±2.4) |
| | Idefics2 - *supervised fine-tuning* | 0.27 | 3.18 (±2.6) |
| | Idefics2 - *supervised fine-tuning + DPO* | **0.46** | **2.40 (±2.6)** |

Table 6.7: Model performance across training approaches. We show in **bold** the best results from the DPO training (epoch 2, see Figure 6.9).

- **Source block prediction:** we randomly select a block from the list of all blocks, ensuring that it is not the same as the gold prediction. Blocks at the bottom of a stack are excluded (e.g., only the top block in a stack can be picked up), as well as the candidate block if the dialogue includes a TPR. This approach ensures the model continues to train on predicting bounding boxes around blocks, while maintaining plausibility for the alternative (rejected) predictions.

- **Target position prediction:** we apply a random translation to the gold bounding box target position along the x and y axes. The translation must be at least 100 pixels on either axis (as bounding boxes are approximately 100 pixels in size), but no more than ±300 pixels. This constraint prevents the rejected prediction from being unrealistically close to or far from the true location.

### *6.6.3 DPO Training and Results*

During training, the model is presented with both *chosen* and *rejected* predictions,and must select the one most likely to be correct, similar to a classification task. We train the model for 3 epochs, starting from the best supervised fine-tuned checkpoint from Section 6.4 with default losses.

We present the results for the DPO training in Table 6.7 and Figure 6.9. Our findings indicate that DPO is very effective at improving the Idefics2 model further, for both instructions are repairs. Notably, the model exhibits substantial improvements in handling repairs, suggesting that our pre-training strategy combined with DPO is particularly effective in distilling the capability to address repairs into the model.


Figure 6.9: DPO training plots for Idefics2 with the BW-R dataset. We observe that the best performing model is reached after epoch 2, and further training deteriorates performance. At epoch 0, the model is equivalent to the supervised fined-tuned checkpoint from the previous section.

## 6.7 Discussion

This chapter explored the challenges of handling repairs with VLMs in mutli-modal dialogues. The ability to process and respond effectively to corrections within a broader conversation context is essential for collaborative AI, yet few works investigate this ability in truly context-dependent scenarios, where the model actively partakes in the dialogue, providing meaningful candidate responses.

We first introduced a new dataset and benchmark, BlockWorld-Repairs, derived from the corpus in Chapter 3 and the Blocks World (Bisk et al., 2016) environment. We refined the collected dialogues, removing unnecessary turns and adapting their format to

align with a visual-grounding task, suited for state-of-the-art large vision and language models. Additionally, we validated these dialogues through an in-person human study to establish a baseline for model performance. This highly ambiguous tabletop manipulation setting requires a fine-grained understanding of spatial references and dialogue to resolve complex multi-modal instructions and their subsequent repairs.

Our findings reveal that two VLMs, Idefics2 and LLaVA, struggle to handle BW-R in a zero-shot setting despite their strong visual capabilities (Laurençon et al., 2024; Liu et al., 2024b). Fine-tuning improves models' handling of either instructions or repairs, but not both simultaneously, even though the overall task, *bounding box prediction*, is the same. In our experiments, we proposed alternative fine-tuning strategies that focus on relevant tokens, excluding incorrect candidate responses by calculating cross-entropy losses solely on user turn or final completion tokens. We demonstrated that models benefit from partially masking some input tokens when learning to process TPRs, achieving stronger performance and more generalisable models on one of the sub-tasks. In these cases, models learn to prioritise relevant information across both instructions and repairs, particularly benefiting their ability to process TPRs. However, this approach requires balance, as excessively masking too many tokens can hinder models when data is limited, as is our case, reducing the training signal available to learn the task.

Finally, our error analysis comparing VLMs and human participants reveals stark contrasts in how they process repairs. All models consistently underperform with dialogues that humans find straightforward, and despite their visual-grounding training to predict objects, these VLMs occasionally fail to identify blocks in images altogether. Notably, the GPT-4o model seemed more aligned with human responses than the others, producing similar outputs. We suspect this stronger alignment is due to RLHF, which we explore in Section 6.6. Our findings indicate that fine-tuning a model with DPO can further distil the capability to handle repairs into a model, highlighting the importance of combining these training approaches.

To effectively coordinate with humans through dialogue, models have to be able to resolve these TPRs, similar to how humans can resolve ambiguous referring expressions even when initial instructions are unclear and require clarification through repairs. This chapter demonstrates that specialised training can benefit models, but there is still a significant gap to reach human communication.

# Chapter 7

# Conclusions and Future Directions

This chapter offers an overview of the thesis and summarises its contributions to the broader field. We discuss the limitations of the research undertaken and provide recommendations for future work to build on these findings.

## 7.1 Contributions and Findings

This thesis explores referential miscommunications and their resolution in multi-modal dialogues for collaborative tasks. Resolving referential expressions in dialogues requires addressing both communicative grounding (Clark, 1996) and the symbol problem (Harnad, 1990) in an integrated approach (Larsson, 2018). The findings of this work are particularly relevant for computational models, especially those integrating vision and language in human dialogue contexts. The rest of this section presents the main findings from each analysis chapter and addresses the research questions proposed in Section 1.1:

- **Chapter 3.** We studied situated instruction understanding within the highly ambiguous Blocks World environment (Bisk et al., 2016). Our literature review identified a lack of truly interactive scenarios for researching human-agent dialogue collaboration, with agents typically following instructions without actively contributing to coordination. To address this, we **collected a dialogue corpus** in which a robot agent contributes candidates to the conversation based on user instructions for moving a block on a table. We designed a setup that elicited highly context-dependent corrections, influenced by factors such as previous actions or where the robot is pointing. This corpus tackles the first research question, ***RQ1: What are the ways that we can elicit miscommunications in multi-modal referential conversations?***, ensuring that these dialogues are grounded both in the dialogue context (e.g., corrections depend on previous candidates and instructions) and the visu-

al/spatial context (e.g., the table layout, relationships between blocks, the robot's position). The resulting interactions feature a bi-directional flow of information between the user and the agent, providing an ideal framework for researching miscommunications in collaborative scenarios. Using this dataset, we explored several modelling approaches for instruction understanding and found that **resolving referring expressions was the main challenge** for our models. Through a simplified RL case study, we showed that **learning to handle corrections is beneficial for models** capable of iteratively refining their internal representations with feedback, particularly when perfect predictions (e.g., action execution) cannot be guaranteed, such as in environments with ambiguity or collaborative tasks.

- **Chapter 4.** To approach learning to resolve referring expressions, we introduced the SIMMC 2.0 dataset of collaborative situated dialogues. As part of the challenge, we designed a coreference resolution model that extracts visual features and then combines them with text, finding that visual information is often too noisy to distinguish similar objects in cluttered scenes. Ambiguity played a central challenge in the SIMMC 2.0 environments, where visually indistinguishable objects with overlapping properties (e.g., two red shirts differing only by their relative positions) made perfect resolution of referring expressions difficult. To address this, we experimented with models to detect ambiguity in referring expressions that fail to uniquely identify their referents. We demonstrated that models heavily exploit language biases to achieve strong performance, even without visual input. Our analysis also revealed that language-only models performed well on SIMMC 2.0 tasks when they exploited privileged information (i.e., object IDs), but struggled in its absence, underscoring the inherent difficulty of the task and highlighting the importance of robust unimodal baselines (Thomason et al., 2019). Regarding the research question, ***RQ2: How can models learn robust multi-modal representations to resolve ambiguous referring expressions?***, our experiments explored various approaches for learning these representations by combining linguistic and visual information from dialogues, as well as with additional details from the SIMMC 2.0 data itself. The findings indicate that **vision is unreliable** in scenes with many similar objects and that our model struggles to resolve coreferences following an ambiguous referring expression in the dialogues.

- **Chapter 5.** We analysed referential ambiguities in SIMMC 2.0 and how users engaged in Clarificational Exchanges CEs to repair miscommunications. We proposed a taxonomy of CEs based on the properties and modalities used to resolve ambiguities, more fine-grained than previous approaches at level 3 of Clark's (1996) joint action ladder. This taxonomy allowed us to inspect the contributions of different modalities in resolving coreferences. We then evaluated several models on their ability to process and integrate repairs in dialogues, revealing differences across model architectures and training regimes. We find that visual information is difficult to leverage effectively, whereas incorporating **additional training objectives** for learning disentangled object representations (Bengio et al., 2013) and explicitly encoding object relationships improves model robustness and handling of CEs. Language-only models did well when leveraging metadata or global object representations, but this was ineffective in the case of CEs. With respect to the research question, ***RQ3: What are the signals necessary for learning to process repairs?***, our results indicate that holistic, object-centric representations encoding attribute and relationship information (Seitzer et al., 2023) are most effective to processing repairs within dialogue contexts,enabling dynamic adaptation based on the interaction. However, this must be balanced with a strong language objective as to not undermine context carry-over across longer dialogue sequences.

- **Chapter 6.** We introduced the BlockWorld-Repairs benchmark, derived from the corpus presented in Chapter 3 and narrowing our focus to Third Position Repairs. We adapted the dialogues to match the format of visual-grounding tasks common with state-of-the-art VLMs. Our proposed benchmark assesses the ability of VLMs to follow complex instructions and repairs in the Blocks World environment. Despite their strong visual grounding capabilities (Laurençon et al., 2024; Liu et al., 2024b), we found that these models struggled in zero-shot settings and showed limited generalisation in handling both instructions and repairs, even after fine-tuning. To address this, we proposed alternative fine-tuning strategies that focus on relevant information in the input text sequences, ultimately improving the models' generalisation capabilities. We demonstrated that **prioritising critical details** benefits models' ability to process third position repairs. Furthermore, we conducted a human study to establish a baseline and compare differences between human and

model capabilities. Our findings firstly show that humans excel at following instructions and yet can further exploit repairs to improve their performance in this highly ambiguous environment. In contrast, we find that **models underperform** on dialogues that study participants found straightforward, and is only through our fine-tuning regimes that models become adept at exploiting repairs. This chapter explored the research question, ***RQ4: How can we distil in models the capability to process repairs?*** whilst heavily building on insights learned from RQ2 and RQ3. Our experiments and analysis revealed the importance of **learning from interactive settings**, focusing on the relevant details of the overarching task so that models learn to extract the correct signals for processing repairs (RQ3). While the incorrect candidates provided by an agent during interaction can aid in learning dialogue coordination, they may also hinder the model's ability to solve the primary task, altering learned representations (RQ2). Our results indicate the need to balance these factors to develop agents capable of coordinating actions through language.

This thesis revolved around a central research question, ***How can models process referential miscommunications in multi-modal collaborative tasks?***, investigating how conversational agents can learn to address the challenges posed by miscommunications and exploit subsequent repairs. Our findings highlight that this under-explored aspect of dialogue requires particular attention if we aim to develop models, or agents, capable of collaborating with humans across diverse tasks. Repairs are not just vital for dialogue coordination (Healey et al., 2018), but should be expected in dynamic contexts, especially since current (and likely future) agents lack a perfect, all-encompassing understanding of the environment and have differing perspectives. The real world is inherently messy and noisy, filled with similar objects and changing conditions where even the slightest changes may complicate the task (e.g., new lighting conditions). Miscommunications and repairs, amongst other dialogue phenomena, enable humans to coordinate their linguistic and non-linguistic actions across diverse situations (Eshghi et al., 2022; Mills, 2014). Equipping conversational agents with these capabilities is thus essential for effective and adaptative interaction. However, as we show several times in this thesis, models particularly struggle with miscommunications and their repairs in dialogues. Unlike humans, who leverage repairs to facilitate collaboration, current models fail to fully exploit them, indicating they cannot yet process interaction in the same nuanced way.

This thesis provides important considerations for designing vision-and-language dialogue models and multi-modal conversational agents. Firstly, models require multi-modal object representations with disentangled attributes (i.e., representing attributes such as colour or shape independently from each other) to effectively reason about the surrounding environment and resolve referential miscommunications. We demonstrate the importance of incorporating additional training objectives for learning disentangled object representations, enabling models to process repairs that integrate new contextual information effectively, as we show here. Finally, we argue for training in interactive settings where the agent or model can make mistakes and subsequently learn from corrections. The ability to adapt to novel or unexpected situations is essential for developing generalisable models (Parekh et al., 2024). This thesis releases both a benchmark and a taxonomy for assessing the robustness of models in handling miscommunications, with a focus on clarificational exchanges and third position repairs in multi-modal collaborative dialogues involving referential ambiguities. Designing interactive settings that mirror the complexities of real world scenarios is crucial for models to learn to deal with noise and ambiguities. Such an approach is fundamental for achieving true collaboration with humans, even in the current state of generative AI.

## 7.2 Limitations of Current Work

Throughout this thesis, we have explored the holistic nature of dialogue interactions, and how the environment or task shapes various aspects of collaboration. Miscommunications and repairs, while common in human dialogues, are poorly represented in publicly available datasets. Many works have attempted to emphasise the unique dynamics of dialogue, distinguishing it from static text sequences and addressing additional considerations, such as interlocutor proximity or miscommunications. Our research focused on the integration of vision and language modalities in situated dialogues, a narrow subset of dialogue interactions. Thus, our findings may be limited in other contexts, where the conversational agent and human do not share the same (physical) space or when dialogues extend over longer contexts (e.g., online chatbot).

This work examines understanding referring expressions and processing their repairs, leaving aside other closely related tasks. Generating clarification requests is important for determining whether a referring expression aligns the model's expectations or inter-

nal representations. This involves detecting ambiguities or insufficient information to distinguish a referent (briefly explored in Chapter 4) and subsequently producing an utterance to clarify a relevant property of the referent. While prior works have explored this aspect extensively (Addlesee et al., 2024; Madureira and Schlangen, 2024; Madureira and Schlangen, 2023; Willemsen and Skantze, 2024; Willemsen et al., 2023; Shi et al., 2022; Zhu et al., 2021; Addlesee and Eshghi, 2021; Gervits et al., 2021; Aliannejadi et al., 2021), we do not address it here. Instead, we assume that the agent has already generated the CR (e.g., CRs from SIMMC 2.0) or has a candidate (e.g., candidate turns in BW-R dialogues Chapter 6).

In Chapter 3, we introduced a dialogue corpus designed to capture the complexities of interaction. However, the dialogues were collected with a rule-based agent within AMTurk as an annotation task, enabling us to scale the collection while focusing on instructions and their corrections. Deploying this type of agent also enabled us to run 'simulated' scenarios to obtain training data for the agent itself, at the cost of a true collaborative interaction. A more ideal approach would involve designing a Wizard-of-Oz or human-human experiment to collect the more intrinsic elements of dialogue interactions, with significantly greater time and cost.

The Blocks World environment used within this thesis is visually simplistic compared to the images or domains commonly used to pretrain VLMs (e.g., photo-realistic images like MSCOCO (Lin et al., 2014a) or simulations like AI2Thor (Kolve et al., 2017)). Similar to works in Embodied AI (e.g., ALFRED (Shridhar et al., 2020); Simbot Arena (Gao et al., 2023b), *inter alia*), we did not attempt to model the full spectrum of robotic manipulation tasks (e.g., Octo Model Team et al., 2024). Instead, we abstracted object manipulation as API call generation, where actions are defined with a bounding box, or *coordinates*, for the object to be manipulated. These high-level actions avoided other robotic research areas, such as motion planning or object grasping.

Our environment also features information asymmetry, where the human has access to the final arrangement whilst the agent does not, in settings with extreme ambiguity. We expect that few human-human conversations would mirror these dialogues, where one participant possesses all the information in a highly ambiguous context. However, our focus was on eliciting miscommunications in dialogue and investigating how to process their repairs. Simplifying the visual complexity allowed us to minimize confounding factors and attend to the fundamental aspects of dialogue coordination in situated, multi-

modal settings, where even larger models with stronger visual representations may not be able to solve without cooperation.

## 7.3 Future Work

This thesis lays the groundwork for researching multi-modal miscommunications with conversational agents. We identify several areas of future work below.

**Conversational Agents Robust to Ambiguity** Our exploration was characterised by scenarios with high levels of ambiguity, where successful outcomes relied on effective dialogue cooperation. Much of our literature review shows that embodied AI benchmarks, such as ALFRED (Shridhar et al., 2020) or Simbot Arena (Gao et al., 2023b), minimise ambiguity to the point that they remove the need for corrections or collaboration, not requiring the ability to establish common ground (Suglia et al., 2024). In these simulated environments, agents typically decompose high-level instructions into smaller steps, with language playing a limited role: setting the initial goal. Any subsequent dialogue becomes redundant as agents' capabilities sufficiently improve (e.g., avoiding visual ambiguities as in PhotoBook (Haber et al., 2019), or 'learning' from pretraining biases that mugs are typically found in kitchens (Padmakumar et al., 2022)). Therefore, future work should design environments where ambiguities and miscommunications are prevalent and cannot be resolved solely through models with superior individual capabilities, i.e., improved object detection or spatial reasoning. Instead, these scenarios should encourage active language cooperation to overcome misunderstandings and establish common ground. This capability is essential for developing generalisable conversational agents and advancing generative AI systems.

**Learning from Interaction** In this thesis, we argue that learning from interactive settings is crucial for developing robust computational models, as dialogue coordination cannot emerge from training on text sentences alone. However, our exploration is limited to relatively simple scenarios, the Blocks World setting (Chapters 3 and 6) and SIMMC 2.0 (Chapters 4 and 5), which involve at most one or two 'collaborative' turns. Future research should investigate learning from richer interactive environments, such as carefully designed language games, which address the limitations of prior approaches.

These games should prioritise highly context-dependent dialogues (in contrast to limited-context approaches, see Agarwal et al. (2020)) and facilitate meaningful information exchange between participants (i.e., beyond simple yes/no questions as in GuessWhat?! (de Vries et al., 2017)) to capture the entire collaborative process (Clark and Wilkes-Gibbs, 1986). Simulated language games, in particular, may offer a promising avenue for learning strategies to improve conversational agent's capabilities on communicative and symbol groundings.

**Learning in Interaction** Also closely related, future work could explore learning within interactive settings, enabling models to dynamically adapt based on feedback or corrections during dialogues. Previous works have demonstrated that models can learn new meanings and concepts from others in interaction (i.e., compositional semantics) (Eshghi et al., 2013; Yu et al., 2016b). For instance, understanding that *"pass my mug"* refers to a specific mug following a brief clarification exchange (*"Which mug? – The blue one"*). While our benchmark and data do not support indefinite training of models, continual learning (Hassabis et al., 2017)[1] offers a promising alternative. This approach enables models to train in more flexible environments, gradually incorporating new tasks and retrospectively learning from signals in past interactions (Chen et al., 2024).

**Understanding Differing Perspectives** Our work focused on processing repairs within dialogues, assuming shared human-agent views (i.e., frontal images), yet it did not explore how to bridge perspectives between conversational agents and users. Humans would temporarily adapt or *align* their perspectives (Dobnik et al., 2020) with other interlocutors during conversations to facilitate coordination and resolve referential expressions. However, this thesis did not explicitly model such alignment, which would be particularly valuable for handling the spatial references explored in our work. We anticipate that a conversational agent capable of reasoning about the divergence between perspectives and dynamically aligning them may be able to cooperate more effectively in a broader range of scenarios.

[1] See Wang et al. (2024) for a comprehensive survey.

# Appendix A

# Supplementary Material for Chapter 5

## A.1 Example Dialogues

We present several dialogues that contain one or more CEs from the SIMMC 2.0 dataset below. We also include their annotations according to our taxonomy from Section 5.4.

| | | |
|---|---|---|
| **User:** | *Hello, do you have any jackets for me to look at?* | |
| **System:** | *Sure, what do you think of the light grey jacket hanging up high at the back left?* | |
| **User:** | *Do you have anything with a similar size range to the black sweater beside the light grey jacket?* | |
| **System:** | *Sorry, I don't have anything similar to that* | |
| **User:** | *What size is that sweater anyways?* [Referential Ambiguity] | **Before-CR** |
| **System:** | *The black one?* [Clarification Request] | **CR** |
| **User:** | *Yes exactly* [Clarification]<br>`Tags in CE:`<br>`- Individual Property (Colour)`<br>`- Dialogue History (Confirmation)`<br>  | **After-CR** |

Table A.1: Sample dialogue 1 with a CE from the SIMMC 2.0 dataset.

| | | |
|---|---|---|
| **User:** | *I need some help picking out a jacket.* | |
| **System:** | *Do you like the look of that solid black one hanging in the bottom row up front on the right?* | |
| **User:** | *I do like the look of that one. Will you please put that in my cart?* | |
| **System:** | *Sure! I'll add that black jacket to your cart right now.* | |
| **User:** | *Now could we look for a grey coat?* | |
| **System:** | *There's a grey coat hanging directly above the black jacket you like.* | |
| **User:** | *Actually first I'd like to know which brand makes that jacket. If you have any data on what other customers think of it, that'd be good to hear too.* [Referential Ambiguity] | **Before-CR** |
| **System:** | *Sorry, which jacket specifically are you curious about?* [Clarification Request] | **CR** |
| **User:** | *The black one up front next to the skirt* [Clarification]<br>`Tags in CE:`<br>`- Individual Property (Type, Colour)`<br>`- Relational Context (Relational)` | **After-CR** |
| **System:** | *That jacket is a Yogi Fit original, and other buyers have rated it a 4.0 on average* | |
| **User:** | *Are there any super ritzy options with stellar reviews?* | |
| **System:** | *I don't see any, sorry. Is there something else I can help you with?* | |
| **User:** | *I'd like to find a dress in extra large.* | |
| **System:** | *I have a feeling you'll like the dresses on this first rack up front and center. Would you wear the purple one on the left, the brown one on the right, or the grey one in front?* | |
| **User:** | *Woops, I didn't even ask you what sizes that jacket came in.* [Referential Ambiguity] | **Before-CR** |
| **System:** | *Which jacket do you want to know the sizes of?* [Clarification Request] | **CR** |
| **User:** | *That black one that I had you put in my cart* [Clarification]<br>`Tags in CE:`<br>`- Individual Property (Type, Colour)`<br>`- Dialogue History (Previous history)`<br>  | **After-CR** |

Table A.2: Sample dialogue 2 with a CE from the SIMMC 2.0 dataset.

# Appendix B

# Supplementary Material for Chapter 6

## B.1 Experimental Setup

### *B.1.1 Prompts*

Each task has its own slightly different prompt, see Table B.1. We also provide example full prompts used during zero-shot and fine-tuning in Table B.2.

| Task | Task Instruction |
|---|---|
| Source Block Selection | Answer only with the bounding box of the block mentioned. The bounding box consists of 4 values between 0 and 1. Here is an example: [0.000, 0.123, 0.075, 0.204] |
| Target Position Prediction | Answer only with the final bounding box location mentioned. The bounding box consists of 4 values between 0 and 1. Here is an example: [0.000, 0.123, 0.075, 0.204] |

Table B.1: Task instructions used to prompt models.

**Parsing Output** When parsing model generations, we try to parse a bounding box with 4 decimal numbers. However, models sometimes generate invalid boxes or additional tokens, particularly in zero-shot settings. In these cases, less than 3%, we count this prediction as 'failed' and use the output of the random baseline to calculate other metrics. Idefics2 and GPT-4o are more prone to generate additional tokens, which we ignore them if the bounding box coordinates can be successfully parsed.

### *B.1.2 Supervised Model Fine-Tuning*

For our experiments, we use the default hyperparameters for each model (Idefics2 and LLaVA). We train for 1 epoch as we did not observe benefits from training for longer. Images had the size of 1024x576 pixels, and we used the model's default image processing pipelines. Training batches had both instruction and repair examples, sampled at random. We did not use custom sampling to balance the type of entries in batches and leave this for future work.

We fine-tune the VLMs with the recommended parameter-efficient methods: LoRA (Hu et al., 2021) for LLaVA and QLoRA (Dettmers et al., 2023) for Idefics2. We used 2x NVIDIA A40 (40GB) at most and set the maximum sequence length to 2048. Each experiment (training and testing) takes around 1 hour to fully complete. The final table of experiments takes approximately 30 GPU hours to run.

### *B.1.3 Object Detection and Task Predictions*

Splitting the task into object detection and then prediction out of a list of candidates did not show any advantages in our initial tests, possibly because these models are already pre-trained with object detection objectives (Liu et al., 2024a; Laurençon et al., 2024). We already see better performance with source block prediction than with target location prediction. Thus, we unified both tasks into a single common task format (bounding box prediction) for easier comparison. Additionally, we would like to point out that target prediction would be a harder task because it requires the model to derive a bounding box for an empty board location, which models particularly struggle as shown in Section 6.5.

## B.2 Additional Error Analysis

We provide in Table B.3 the results of selecting GPT-4o performance as the proxy for example difficulty. Since we only have 1 prediction value for each BlockWorld-Repairs entry and we cannot do the average as with the human annotations, we instead order the data by distance (source and target separately) and divide it into three equal sizes at the 33rd and 66th percentiles: 1) **Easy** category contains the top 33% of entries (shortest distance to source/target); 2) **Medium** contains entries that are between the 33% and 66%

distances; and 3) **Hard** is reserved for the worst performing 33% entries.

We find that Idefics2 and LLaVA have low performance on all levels, whilst GPT-4o only surpasses humans on easy dialogues (not surprising), but quickly deteriorates to distances worse than random on hard dialogues. Interestingly, humans perform well across all difficulties except the hard target predictions. We also observe that the mean word trends get stronger (Table B.4), although we leave the analysis to prove a correlation for future work. These results further show that models do not process TPRs as consistently as humans.

As also discussed in Section 6.5, it seems that easier referring expressions or TPRs have a lower number of words on average than harder ones, even when using GPT-4o as the proxy for example difficulty, see Table B.4.

| Model | Sample |
|---|---|
| Idefics2 8B | User: <image> Answer only with the bounding box of the block mentioned. The bounding box consists of 4 values between 0 and 1. Here is an example: [0.000, 0.123, 0.075, 0.204]. take the block in the very middle and move it slightl left and down.<end_of_utterance><br>Assistant: location bounding box [0.329, 0.724, 0.339, 0.741] I recall you wanting the block here, is this correct?<end_of_utterance><br>User: move it to the right a little<end_of_utterance><br>Assistant:<br>`Model generation:` location bounding box [0.394, 0.585, 0.436, 0.682] |
| LLaVA v1.5 7B | <s> A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. Answer only with the bounding box of the block mentioned. The bounding box consists of 4 values between 0 and 1. Here is an example: [0.000, 0.123, 0.075, 0.204].<br>USER: <image> There is a block in the center of the board. Take that block and move it to the left in front of the other block.<br>ASSISTANT: block bounding box block bounding box [0.391, 0.635, 0.434, 0.735] Can you confirm it is this block?</s><br>USER: The block that is a row closer to you and to the right.<br>ASSISTANT:<br>`Model generation:` block bounding box [0.391, 0.635, 0.434, 0.735] |
| GPT-4o | System: You are a helpful assistant that responds with only with the bounding box of the location mentioned. The bounding box consists of 4 values between 0 and 1. Here is an example: [0.000, 0.123, 0.075, 0.204].<br>User: <image> Take the block that is furthest to the right in the fourth row and place it in front of all the blocks on the table.<br>Assistant: block bounding box [0.749, 0.714, 0.800, 0.818] Is it this block?<br>User: Take the block that is directly above this block and to the left of it.<br>`Model generation:` block bounding box [0.624, 0.571, 0.675, 0.675] |

Table B.2: Full prompts given to models, including special tokens.

| Difficulty | Model | Source | | Target |
|---|---|---|---|---|
| | | Acc ↑ | Mean (SD) ↓ | Mean (SD) ↓ |
| Easy | Idefics2 | 0.43 | 2.60 (±2.6) | 6.11 (±2.9) |
| | LLaVA | 0.48 | 2.02 (±2.0) | 4.20 (±2.9) |
| | Humans | 0.78 | 1.07 (±1.1) | 2.05 (±2.1) |
| | GPT-4o | 0.93 | .00 (±.0) | 1.27 (±.5) |
| Medium | Idefics2 | 0.47 | 2.29 (±2.3) | 4.97 (±2.6) |
| | LLaVA | 0.47 | 2.01 (±2.0) | 4.25 (±2.7) |
| | Humans | 0.77 | 1.05 (±1.1) | 2.23 (±2.3) |
| | GPT-4o | 0.58 | 1.26 (±1.3) | 2.63 (±.4) |
| Hard | Idefics2 | 0.04 | 4.91 (±4.9) | 5.92 (±3.1) |
| | LLaVA | 0.22 | 4.61 (±4.6) | 5.69 (±3.4) |
| | Humans | 0.64 | 2.52 (±2.5) | 4.84 (±3.7) |
| | GPT-4o | 0.00 | 8.33 (±8.3) | 6.31 (±3.1) |

Table B.3: Model performances across subsets with increasing levels of complexity (*easy, medium, hard*) using GPT-4o performance as the difficulty proxy.

| Difficulty Level | Source Block | Target Position |
|---|---|---|
| *Human Performance* | | |
| Easy | 28.69 (±12.20) | 32.19 (±14.66) |
| Medium | 28.80 (±12.97) | 31.27 (±12.86) |
| Hard | 27.62 (±12.19) | 27.46 (±12.35) |
| *GPT-4o Performance* | | |
| Easy | 29.95 (±12.09) | 34.29 (±13.81) |
| Medium | 29.62 (±12.02) | 31.46 (±12.45) |
| Hard | 26.52 (±12.31) | 24.54 (±11.73) |

Table B.4: Mean and Standard Deviation (SD) for test dialogues in BW-R by difficulty level and task. We differentiate between using human performance (Section 6.5) and GPT-4o performance (Appendix B.2) as the difficulty proxies.

# References

Addlesee, Angus and Arash Eshghi (2021). Incremental Graph-Based Semantics and Reasoning for Conversational AI. In: *Proceedings of the Reasoning and Interaction Conference (ReInAct 2021)*. Ed. by Christine Howes, Simon Dobnik, Ellen Breitholtz, and Stergios Chatzikyriakidis. Gothenburg, Sweden: Association for Computational Linguistics, pp. 1–7.

Addlesee, Angus, Oliver Lemon, and Arash Eshghi (2024). Clarifying Completions: Evaluating How LLMs Respond to Incomplete Questions. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue. Torino, Italia: ELRA and ICCL, pp. 3242–3249.

Agarwal, Shubham, Trung Bui, Joon-Young Lee, Ioannis Konstas, and Verena Rieser (2020). History for Visual Dialog: Do we really need it? In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 8182–8197.

Albert, Saul and J. P. de Ruiter (2018). Repair: The Interface Between Interaction and Cognition. In: *Topics in Cognitive Science* 10.2, pp. 279–313.

Aliannejadi, Mohammad, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev (2021). Building and Evaluating Open-Domain Dialogue Corpora with Clarifying Questions. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4473–4484.

Allwood, Jens (1995). An Activity Based Approach to Pragmatics. In: *Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics*. Ed. by Harry Bunt and William Black. John Benjamins, pp. 47–80.

Anderson, Anne H., Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry S. Thompson, and Regina Weinert (1991). The Hcrc Map Task Corpus. In: *Language and Speech* 34.4, pp. 351–366.

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018a). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In: *CVPR*.

Anderson, Peter, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel (2018b). Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683.

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh (2015). VQA: Visual Question Answering. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. USA: IEEE Computer Society, pp. 2425–2433.

Appelgren, Mattias (2022). Interactive task learning from corrective feedback. PhD thesis. The University of Edinburgh.

Artzi, Yoav and Luke Zettlemoyer (2013). Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. In: *Transactions of the Association for Computational Linguistics* 1. Ed. by Dekang Lin and Michael Collins, pp. 49–62.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). Layer Normalization.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In: *Proceedings of the 3rd International Conference on Learning Representations*. ICLR 2015. San Diago, CA, USA.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan (2022a). Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback.

Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan (2022b). Constitutional AI: Harmlessness from AI Feedback.

Balaraman, Vevake, Arash Eshghi, Ioannis Konstas, and Ioannis Papaioannou (2023). No that‘s not what I meant: Handling Third Position Repair in Conversational Question Answering. In: *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani. Prague, Czechia: Association for Computational Linguistics, pp. 562–571.

Barsalou, Lawrence W. (1999). Perceptual symbol systems. In: *Behavioral and Brain Sciences* 22.4, pp. 577–660.

Barsalou, Lawrence W. (2008). Grounded Cognition. In: *Annual Review of Psychology* 59.1, pp. 617–645.

Bartneck, Christoph, Dana Kulić, Elizabeth Croft, and Susana Zoghbi (2009). Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots. In: *International Journal of Social Robotics* 1.1, pp. 71–81.

Bender, Emily M. and Alexander Koller (2020). Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 5185–5198.

Bengio, Y., P. Simard, and P. Frasconi (1994). Learning long-term dependencies with gradient descent is difficult. In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). Representation learning: A review and new perspectives. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin (2003). A neural probabilistic language model. In: *J. Mach. Learn. Res.* 3.null, pp. 1137–1155.

Bengio, Yoshua and Yann LeCun (2007). Scaling Learning Algorithms toward AI. In: *Large-Scale Kernel Machines*. The MIT Press.

Benotti, Luciana and Patrick Blackburn (2021). A recipe for annotating grounded clarifications. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, pp. 4065–4077.

Bird, Steven, Ewan Klein, and Edward Loper (2009). Natural language processing with python. en. Sebastopol, CA: O'Reilly Media.

Bisk, Yonatan, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian (2020). Experience Grounds Language. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 8718–8735.

Bisk, Yonatan, Kevin J. Shih, Yejin Choi, and Daniel Marcu (2018). Learning interpretable spatial operations in a rich 3D blocks world. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press.

Bisk, Yonatan, Deniz Yuret, and Daniel Marcu (2016). Natural Language Communication with Robots. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, pp. 751–761.

Blukis, Valts, Dipendra Misra, Ross A. Knepper, and Yoav Artzi (2018). Mapping Navigation Instructions to Continuous Control Actions with Position-Visitation Prediction. In: *Proceedings of the Conference on Robot Learning (CORL)*. Zurich, Switzerland.

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba (2016). OpenAI Gym.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher

Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). Language Models Are Few-Shot Learners. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc.

Buß, Okko and David Schlangen (2011). DIUM : An Incremental Dialogue Manager That Can Produce Self-Corrections. In: *Proceedings of SemDial 2011 (Los Angelogue), Los Angeles, CA*, pp. 47–54.

Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). End-to-End Object Detection with Transformers. In: *Computer Vision – ECCV 2020*. Springer International Publishing, pp. 213–229.

Chai, Joyce Y. (2018). Language to Action: Towards Interactive Task Learning with Physical Agents. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, p. 6.

Chen, David L. and Raymond J. Mooney (2011). Learning to interpret natural language navigation instructions from observations. In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* 1, pp. 859–865.

Chen, Howard, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi (2019). TOUCHDOWN: Natural Language Navigation and Spatial Reasoning in Visual Street Environments. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Keqin, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao (2023). Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. In: *arXiv preprint arXiv:2306.15195*.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020a). A simple framework for contrastive learning of visual representations. In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org.

Chen, Yen-Chun, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2020b). Uniter: Universal image-text representation learning. In: *ECCV*.

Chen, Zizhao, Mustafa Omer Gul, Yiwei Chen, Gloria Geng, Anne Wu, and Yoav Artzi (2024). Retrospective Learning from Interactions.

Chi, Ta-Chung, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-tur (2020). Just Ask: An Interactive Learning Framework for Vision and Language Navigation. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.03, pp. 2459–2466.

Chiang, Wei-Lin, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing (2023). Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chiyah Garcia, Francisco J., David A. Robb, Atanas Laskov, Xingkun Liu, Pedro Patron, and Helen Hastie (2018a). Explainable Autonomy: A Study of Explanation Styles for Building Clear Mental Models. In: *Proceedings of The 11th International Natural Language Generation Conference*. INLG'18. Tilburg, The Netherlands: ACM, pp. 99–108.

Chiyah Garcia, Francisco Javier, José Lopes, and Helen Hastie (2020a). Natural Language Interaction to Facilitate Mental Models of Remote Robots. In: *Workshop on Mental Models of Robots*. HRI'20. Cambridge, UK: ACM.

Chiyah Garcia, Francisco Javier, José Lopes, Xingkun Liu, and Helen Hastie (2020b). CRWIZ: A Framework for Crowdsourcing Real-Time Wizard-of-Oz Dialogues. eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed.

by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Marseille, France: European Language Resources Association, pp. 288–297.

Chiyah Garcia, Francisco Javier, David A. Robb, Xingkun Liu, Atanas Laskov, Pedro Patron, and Helen Hastie (2018b). Explainable Autonomy: A Study of Explanation Styles for Building Clear Mental Models. In: *Proceedings of the 11th International Conference on Natural Language Generation*. Ed. by Emiel Krahmer, Albert Gatt, and Martijn Goudbeek. Tilburg University, The Netherlands: Association for Computational Linguistics, pp. 99–108.

Chiyah Garcia, Francisco Javier, Simón C. Smith, José Lopes, Subramanian Ramamoorthy, and Helen Hastie (2021). Self-Explainable Robots in Remote Environments. In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '21 Companion. Boulder, CO, USA: Association for Computing Machinery, pp. 662–664.

Chiyah-Garcia, Javier, Prasoon Goyal, Michael Johnston, and Reza Ghanadan (2024a). Adapting LLM Predictions in In-Context Learning with Data Priors. In: *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*. Ed. by Sachin Kumar, Vidhisha Balachandran, Chan Young Park, Weijia Shi, Shirley Anugrah Hayati, Yulia Tsvetkov, Noah Smith, Hannaneh Hajishirzi, Dongyeop Kang, and David Jurgens. Miami, Florida, USA: Association for Computational Linguistics, pp. 305–316.

Chiyah-Garcia, Javier, Alessandro Suglia, and Arash Eshghi (2024b). Repairs in a Block World: A New Benchmark for Handling User Corrections with Multi-Modal Language Models. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 11523–11542.

Chiyah-Garcia, Javier, Alessandro Suglia, Arash Eshghi, and Helen Hastie (2023). 'What are you referring to?' Evaluating the Ability of Multi-Modal Dialogue Models to Process Clarificational Exchanges. In: *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani. Prague, Czechia: Association for Computational Linguistics, pp. 175–182.

Chiyah-Garcia, Javier, Alessandro Suglia, José David Lopes, Arash Eshghi, and Helen Hastie (2022). Exploring Multi-Modal Representations for Ambiguity Detection & Coreference Resolution in the SIMMC 2.0 Challenge. In: *AAAI 2022 DSTC10 Workshop*.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel (2022). PaLM: Scaling Language Modeling with Pathways.

Cirik, Volkan, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick (2018). Visual Referring Expression Recognition: What Do Systems Actually Learn? In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, pp. 781–787.

Clark, Herbert H. (1996). Using Language. Cambridge University Press.

Clark, Herbert H. and Susan E Brennan (1991). Grounding in communication. In: *Perspectives on socially shared cognition.* American Psychological Association, pp. 127–149.

Clark, Herbert H. and C. R. Marshall (1981). Definite reference and mutual knowledge. In: *Elements of discourse understanding*. Cambridge: Cambridge University Press.

Clark, Herbert H. and E. A. Schaefer (1989). Contributing to discourse. In: *Cognitive Science* 13, pp. 259–294.

Clark, Herbert H. and Deanna Wilkes-Gibbs (1986). Referring as a Collaborative Process. In: *Cognition* 22, pp. 1–39.

Colman, M. and P. G. T. Healey (2011). The Distribution of Repair in Dialogue. In: *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*. Boston, MA, pp. 1563–1568.

Curry, Amanda Cercas, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xu Xinnuo, Ondrej Dusek, Arash Eshghi, Ioannis Konstas, Verena Rieser, and Oliver Lemon (2018). Alana v2: Entertaining and Informative Open-domain Social Dialogue using Ontologies and Entity Linking. English. In: *1st Proceedings of Alexa Prize (Alexa Prize 2018)*.

Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8609–8613.

Dale, Robert and Ehud Reiter (1995). Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. In: *Cognitive Science* 19.2, pp. 233–263.

Dan, Soham, Michael Zhou, and Dan Roth (2021). Generalization in Instruction Following Systems. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, pp. 976–981.

Das, Abhishek, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra (2017). Visual Dialog. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 326–335.

De Pellegrin, Emanuele and Ronald PA Petrick (2024). Planning Domain Simulation: An Interactive System for Plan Visualisation. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 34, pp. 133–141.

de Vries, Harm, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela (2018). Talk the Walk: Navigating New York City through Grounded Dialogue.

de Vries, Harm, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville (2017). GuessWhat?! Visual Object Discovery Through Multi-Modal Dialogue. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dettmers, Tim, Mike Lewis, Younes Belkada, and Luke Zettlemoyer (2022a). LLM.int8(): 8-bit matrix multiplication for transformers at scale. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc.

Dettmers, Tim, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer (2022b). 8-bit Optimizers via Block-wise Quantization. In: *International Conference on Learning Representations*.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer (2023). QLORA: efficient finetuning of quantized LLMs. In: *Proceedings of the 37th International*

*Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186.

Dingemanse, Mark, Seán G. Roberts, Julija Baranova, Joe Blythe, Paul Drew, Simeon Floyd, Rosa S. Gisladottir, Kobin H. Kendrick, Stephen C. Levinson, Elizabeth Manrique, Giovanni Rossi, and N. J. Enfield (2015). Universal Principles in the Repair of Communication Problems. In: *PLOS ONE* 10.9, pp. 1–15.

Dobnik, Simon, Mehdi Ghanimifard, and John Kelleher (2018). Exploring the Functional and Geometric Bias of Spatial Relations Using Neural Language Models. In: *Proceedings of the First International Workshop on Spatial Language Understanding*. Ed. by Parisa Kordjamshidi, Archna Bhatia, James Pustejovsky, and Marie-Francine Moens. New Orleans: Association for Computational Linguistics, pp. 1–11.

Dobnik, Simon, Christine Howes, and John Kelleher (2015). Changing perspective: Local alignment of reference frames in dialogue. In: *Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*. Gothenburg, Sweden: SEMDIAL.

Dobnik, Simon, John D. Kelleher, and Christine Howes (2020). Local Alignment of Frame of Reference Assignment in English and Swedish Dialogue. In: *Spatial Cognition XII*. Springer International Publishing, pp. 251–267.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). An Image is Worth 16x16

Words: Transformers for Image Recognition at Scale. In: *International Conference on Learning Representations*.

Dubey, Abhimanyu et al. (2024). The Llama 3 Herd of Models.

Enfield, N. J., Mark Dingemanse, Julija Baranova, Joe Blythe, Penelope Brown, Tyko Dirksmeyer, Paul Drew, Simeon Floyd, Sonja Gipper, Rósa Gísladóttir, Gertie Hoymann, Kobin H. Kendrick, Stephen C. Levinson, Lilla Magyari, Elizabeth Manrique, Giovanni Rossi, Lila San Roque, and Francisco Torreira (2013). Huh? What? –A first survey in twenty-one languages. In: *Conversational Repair and Human Understanding*. Ed. by Makoto Hayashi, Geoffrey Raymond, and Jack Sidnell. Cambridge: Cambridge University Press, pp. 343–380.

Eshghi, Arash and Arash Ashrafzadeh (2023). Learning to generate and corr- uh I mean repair language in real-time. In: *Proceedings of SemDial'23 (MariLogue)*.

Eshghi, Arash, Julian Hough, and Matthew Purver (2013). Incremental Grammar Induction from Child-Directed Dialogue Utterances. In: *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*. Ed. by Vera Demberg and Roger Levy. Sofia, Bulgaria: Association for Computational Linguistics, pp. 94–103.

Eshghi, Arash, Christine Howes, and Eleni Gregoromichelaki (2022). Action coordination and learning in dialogue. In: *Probabilistic Approaches to Linguistic Theory*. Ed. by J-P Bernardy, R Blanck, S Chatzikyriakidis, S Lappin, and A Maskharashvili. Probabilistic Approaches to Linguistic Theory. CSLI Publications, pp. 361–422.

Ferreira, Victor S. (2008). Ambiguity, Accessibility, and a Division of Labor for Communicative Success. In: *Advances in Research and Theory*. Elsevier, pp. 209–246.

Frisson, Steven (2009). Semantic Underspecification in Language Processing. In: *Language and Linguistics Compass* 3.1, pp. 111–127.

Gao, Jianfeng, Michel Galley, and Lihong Li (2019). Neural Approaches to Conversational AI. In: *Foundations and Trends in Information Retrieval*.

Gao, Leo, John Schulman, and Jacob Hilton (2023a). Scaling Laws for Reward Model Overoptimization. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 10835–10866.

Gao, Qiaozi, Govind Thattai, Suhaila Shakiah, Xiaofeng Gao, Shreyas Pansare, Vasu Sharma, Gaurav Sukhatme, Hangjie Shi, Bofei Yang, Desheng Zhang, Lucy Hu, Karthika Arumugam, Shui Hu, Matthew Wen, Dinakar Guthy, Shunan Chung, Rohan Khanna, Osman Ipek, Leslie Ball, Kate Bland, Heather Rocker, Michael Johnston, Reza Ghanadan, Dilek Hakkani-Tur, and Prem Natarajan (2023b). Alexa Arena: A User-Centric Interactive Platform for Embodied AI. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., pp. 19170–19194.

Gao, Xiaofeng, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme (2022). DialFRED: Dialogue-Enabled Agents for Embodied Instruction Following. In: *IEEE Robotics and Automation Letters* 7.4, pp. 10049–10056.

Gervits, Felix, Gordon Briggs, Antonio Roque, Genki A. Kadomatsu, Dean Thurston, Matthias Scheutz, and Matthew Marge (2021). Decision-Theoretic Question Generation for Situated Reference Resolution: An Empirical Study and Computational Model. In: *Proceedings of the 2021 International Conference on Multimodal Interaction*. ICMI '21. Montréal, QC, Canada: Association for Computing Machinery, pp. 150–158.

Ginzburg, Jonathan (2012). The Interactive Stance: Meaning for Conversation. Oxford, UK: Oxford University Press.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: *Proceedings*

*of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '14. USA: IEEE Computer Society, pp. 580–587.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). Deep Learning. `http://www.deeplearningbook.org`. MIT Press.

Goodwin, Charles (1981). Conversational organization: Interaction between speakers and hearers. New York: Academic Press.

Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2017). Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6325–6334.

Grice, H. Paul (1969). Utterer's Meaning and Intentions. In: *The Philosophical Review* 68, pp. 147–177.

— (1975). Logic and Conversation. In: *Speech Acts*. BRILL, pp. 41–58.

Gu, Jing, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang (2022). Vision-and-Language Navigation: A Survey of Tasks, Methods, and Future Directions. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 7606–7623.

Haber, Janosch, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández (2019). The PhotoBook Dataset: Building Common Ground through Visually-Grounded Dialogue. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 1895–1910.

Harnad, Stevan (1990). The symbol grounding problem. In: *Physica D: Nonlinear Phenomena* 42.1-3, pp. 335–346.

Hassabis, Demis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick (2017). Neuroscience-Inspired Artificial Intelligence. In: *Neuron* 95.2, pp. 245–258.

Hastie, Helen, Francisco J. Chiyah Garcia, David A. Robb, Pedro Patron, and Atanas Laskov (2017). MIRIAM: A Multimodal Chat-Based Interface for Autonomous Systems. In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ICMI'17. Glasgow, UK: ACM, pp. 495–496.

Hastie, Helen, Katrin S. Lohan, Mike J. Chantler, David A. Robb, Subramanian Ramamoorthy, Ron Petrick, Sethu Vijayakumar, and David Lane (2018). The ORCA Hub: Explainable Offshore Robotics through Intelligent Interfaces. In: *Proceedings of Explainable Robotic Systems Workshop, HRI'18*. Chicago, IL, USA.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Healey, P. G. T., M. Colman, and M. Thirlwell (2005). Analysing Multi-Modal Communication: Repair-Based Measures of Human Communicative Co-ordination. In: *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*. Ed. by J. van Kuppevelt, L. Dybkjaer, and N. Bernsen. Vol. 30. Text, Speech and Language Technology. Dordrecht: Kluwer, pp. 113–129.

Healey, Patrick G. T., Gregory J. Mills, Arash Eshghi, and Christine Howes (2018). Running Repairs: Coordinating Meaning in Dialogue. In: *Topics in Cognitive Science (topiCS)* 10.2.

Hemanthage, Bhathiya and Oliver Lemon (2022). Global-Local Information-aware Multimodal grounding with GPT for Co-reference Resolution. In: *AAAI 2022 DSTC10 Workshop*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). Long Short-Term Memory. In: *Neural Computation* 9.8, pp. 1735–1780.

Hong, Yining, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan (2023). 3D-LLM: Injecting the 3D World into Large Language Models. In: *arXiv*.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd (2020). spaCy: Industrial-strength Natural Language Processing in Python. In.

Hough, Julian (2015). Modelling Incremental Self-Repair Processing in Dialogue. PhD thesis. Queen Mary University of London.

Hough, Julian and Matthew Purver (2012). Processing Self-Repairs in an Incremental Type-Theoretic Dialogue System. In: *Proceedings of the 16th SemDial Workshop on the Semantics and Pragmatics of Dialogue (SeineDial)*. Paris, France, pp. 136–144.

Hough, Julian and David Schlangen (2015). Recurrent neural networks for incremental disfluency detection. In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pp. 849–853.

— (2016). Investigating Fluidity for Human-Robot Interaction with Real-time, Real-world Grounding Strategies. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Raquel Fernandez, Wolfgang Minker, Giuseppe Carenini, Ryuichiro Higashinaka, Ron Artstein, and Alesia Gainer. Los Angeles: Association for Computational Linguistics, pp. 288–298.

— (2017). It's Not What You Do, It's How You Do It: Grounding Uncertainty for a Simple Robot. In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '17. Vienna, Austria: Association for Computing Machinery, pp. 274–282.

Hu, Edward J, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen (2021). Lora: Low-rank adaptation of large language models. In: *arXiv preprint arXiv:2106.09685*.

Hu, Ronghang, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell (2015). Natural Language Object Retrieval. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4555–4564.

Huang, Yichen, Yuchen Wang, and Yik-Cheung Tam (2022). UNITER-Based Situated Coreference Resolution with Rich Multimodal Input. In: *AAAI 2022 DSTC10 Workshop*.

Huerta-Enochian, Mathew and Seung Yong Ko (2024). Instruction Fine-Tuning: Does Prompt Loss Matter? In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 22771–22795.

Ilinykh, Nikolai and Simon Dobnik (2021a). How Vision Affects Language: Comparing Masked Self-Attention in Uni-Modal and Multi-Modal Transformer. In: *Proceedings of the 1st Workshop on Multimodal Semantic Representations (MMSR)*. Ed. by Lucia Donatelli, Nikhil Krishnaswamy, Kenneth Lai, and James Pustejovsky. Groningen, Netherlands (Online): Association for Computational Linguistics, pp. 45–55.

— (2021b). What Does a Language-And-Vision Transformer See: The Impact of Semantic Information on Visual Representations. In: *Frontiers in Artificial Intelligence* 4.

— (2022). Attention as Grounding: Exploring Textual and Cross-Modal Attention on Entities and Relations in Language-and-Vision Transformer. In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 4062–4073.

Ilinykh, Nikolai, Yasmeen Emampoor, and Simon Dobnik (2022). Look and Answer the Question: On the Role of Vision in Embodied Question Answering. In: *Proceedings of the 15th International Conference on Natural Language Generation*. Ed. by

Samira Shaikh, Thiago Ferreira, and Amanda Stent. Waterville, Maine, USA and virtual meeting: Association for Computational Linguistics, pp. 236–245.

Ji, Heng and Ralph Grishman (2011). Knowledge Base Population: Successful Approaches and Challenges. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, Oregon, USA: Association for Computational Linguistics, pp. 1148–1158.

Jiang, A. Q., A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed (2023a). Mistral 7b. Tech. rep. Technical Report. Research Institution.

Jiang, Yunfan, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan (2023b). VIMA: Robot Manipulation with Multimodal Prompts. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 14975–15022.

Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick (2017). CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997.

Jurafsky, Daniel and James H. Martin (2000). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 1st. USA: Prentice Hall PTR.

— (2024). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd. Online manuscript released August 20, 2024.

Katsakioris, Miltiadis Marios, Ioannis Konstas, Pierre Yves Mignotte, and Helen Hastie (2021). Learning to Read Maps: Understanding Natural Language Instructions from Unseen Maps. In: *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*. Ed. by Malihe Alikhani, Valts Blukis, Parisa Kordjamshidi, Aishwarya Padmakumar, and Hao Tan. Online: Association for Computational Linguistics, pp. 11–21.

Kazemzadeh, Sahar, Vicente Ordonez, Mark Matten, and Tamara Berg (2014). ReferItGame: Referring to Objects in Photographs of Natural Scenes. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 787–798.

Kelleher, John D. and Geert-Jan M. Kruijff (2006). Incremental Generation of Spatial Referring Expressions in Situated Dialog. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Ed. by Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle. Sydney, Australia: Association for Computational Linguistics, pp. 1041–1048.

Kennington, Casey, Livia Dia, and David Schlangen (2015). A Discriminative Model for Perceptually-Grounded Incremental Reference Resolution. In: *Proceedings of the 11th International Conference on Computational Semantics*. Ed. by Matthew Purver, Mehrnoosh Sadrzadeh, and Matthew Stone. London, UK: Association for Computational Linguistics, pp. 195–205.

Kim, Jin-Hwa, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh (2019). CoDraw: Collaborative Drawing as a Testbed for Grounded Goal-driven Communication. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 6495–6513.

Kingma, Diederik and Jimmy Ba (2015). Adam: A Method for Stochastic Optimization. In: *International Conference on Learning Representations (ICLR)*. San Diego, CA, USA.

Kiseleva, Julia, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, Arthur Szlam, Yuxuan Sun, Katja Hofmann, Marc-Alexandre Côté, Ahmed Awadallah, Linar Abdrazakov, Igor Churin, Putra Manggala, Kata Naszadi, Michiel van der Meer, and Taewoon Kim (2022). Interactive Grounded Language Understanding in a Collaborative Environment: IGLU 2021. In: *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*. Ed. by Douwe Kiela, Marco Ciccone, and Barbara Caputo. Vol. 176. Proceedings of Machine Learning Research. PMLR, pp. 146–161.

Kolve, Eric, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. (2017). Ai2-thor: An interactive 3d environment for visual ai. In: *arXiv preprint arXiv:1712.05474*.

Kottur, Satwik, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi (2021). SIMMC 2.0: A Task-oriented Dialog Dataset for Immersive Multimodal Conversations. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4903–4912.

Kottur, Satwik, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach (2019). CLEVR-Dialog: A Diagnostic Dataset for Multi-Round Reasoning in Visual Dialog. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 582–595.

Krahmer, Emiel and Kees van Deemter (2019). Computational Generation of Referring Expressions: An Updated Survey. Ed. by Jeanette Gundel and Barbara Abbott.

Kruijff, Geert-Jan M (2012). There is no common ground in human-robot interaction. In: *Proceedings of SemDial 2012 (SeineDial): The 16th Workshop on the Semantics and Pragmatics of Dialogue*. Citeseer, p. 80.

Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar (2023). CLAM: Selective Clarification for Ambiguous Questions with Generative Language Models.

Lambert, Nathan, Louis Castricato, Leandro von Werra, and Alex Havrilla (2022). Illustrating Reinforcement Learning from Human Feedback (RLHF). In: *Hugging Face Blog*. https://huggingface.co/blog/rlhf.

Larsson, Staffan (2018). Grounding as a Side-Effect of Grounding. In: *Topics in Cognitive Science (topiCS)* 10.2.

Laurençon, Hugo, Léo Tronchon, Matthieu Cord, and Victor Sanh (2024). What matters when building vision-language models?

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (1989). Backpropagation Applied to Handwritten Zip Code Recognition. In: *Neural Computation* 1.4, pp. 541–551.

Lecun, Yann and Yoshua Bengio (1995). Convolutional Networks for Images, Speech and Time Series. In: *The Handbook of Brain Theory and Neural Networks*. Ed. by Michael A. Arbib. The MIT Press, pp. 255–258.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). Deep learning. In: *Nature* 521.7553, pp. 436–444.

Lee, Haeju, Oh Joon Kwon, Yunseon Choi, Minho Park, Ran Han, Yoonhyung Kim, Jinhyeon Kim, Youngjune Lee, Haebin Shin, Kangwook Lee, and Kee-Eung Kim (2022). Learning to Embed Multi-Modal Contexts for Situated Conversational Agents. In:

*Findings of the Association for Computational Linguistics: NAACL 2022*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, pp. 813–830.

Levinson, Stephen (2000). Presumptive Meanings. Cambridge, MA: MIT Press.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 7871–7880.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick (2014a). Microsoft COCO: Common Objects in Context. In: *ECCV*. European Conference on Computer Vision.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014b). Microsoft COCO: Common objects in context. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, pp. 740–755.

Lin, Zhouhan, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio (2017). A structured self-attentive sentence embedding. In: *International Conference on Learning Representations*. ICLR 2017.

Liu, Haotian, Chunyuan Li, Yuheng Li, and Yong Jae Lee (2024a). Improved Baselines with Visual Instruction Tuning. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296.

Liu, Haotian, Chunyuan Li, Qingyang Wu, and Yong Jae Lee (2024b). Visual instruction tuning. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2023). Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. In: *ACM Comput. Surv.* 55.9.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). Roberta: A robustly optimized bert pretraining approach. In: *arXiv preprint arXiv:1907.11692*.

Locatello, Francesco, Stefan Bauer, Mario Lučić, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Frederic Bachem (2019). Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In: *International Conference on Machine Learning*. Best Paper Award.

Lopes, José, Francisco Javier Chiyah Garcia, and Helen Hastie (2020). The Lab vs The Crowd: An Investigation into Data Quality for Neural Dialogue Models. In: *Workshop on Human in the Loop Dialogue Systems at NeurIPS 2020*.

Lopes, José, David A. Robb, Muneeb Ahmad, Xingkun Liu, Katrin Lohan, and Helen Hastie (2019). Towards a Conversational Agent for Remote Robot-Human Teaming. In: *ACM/IEEE International Conference on Human-Robot Interaction*. Vol. 2019-March. IEEE, pp. 548–549.

Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Lu, Xiaofei (2012). The Relationship of Lexical Richness to the Quality of ESL Learners' Oral Narratives. In: *The Modern Language Journal* 96.2, pp. 190–208.

Madureira, Brielen and David Schlangen (2023). Instruction Clarification Requests in Multimodal Collaborative Dialogue Games: Tasks, and an Analysis of the CoDraw Dataset. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Ed. by Andreas Vlachos and Isabelle Au-

genstein. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 2303–2319.

— (2024). Taking Action Towards Graceful Interaction: The Effects of Performing Actions on Modelling Policies for Instruction Clarification Requests. In: *Proceedings of the Third Workshop on Understanding Implicit and Underspecified Language*. Ed. by Valentina Pyatkin, Daniel Fried, Elias Stengel-Eskin, Alisa Liu, and Sandro Pezzelle. Malta: Association for Computational Linguistics, pp. 1–21.

Mao, Junhua, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy (2016). Generation and Comprehension of Unambiguous Object Descriptions. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11–20.

Marge, Matthew and Alexander Rudnicky (2015). Miscommunication Recovery in Physically Situated Dialogue. In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Alexander Koller, Gabriel Skantze, Filip Jurcicek, Masahiro Araki, and Carolyn Penstein Rose. Prague, Czech Republic: Association for Computational Linguistics, pp. 22–31.

McCallum, Sabrina, Max Taylor-Davies, Stefano Albrecht, and Alessandro Suglia (2023). Is feedback all you need? Leveraging natural language feedback in goal-conditioned RL. In: *NeurIPS 2023 Workshop on Goal-Conditioned Reinforcement Learning*.

Mehta, Nikhil and Dan Goldwasser (2019). Improving Natural Language Interaction with Robots Using Advice. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1962–1967.

Mei, Hongyuan, Mohit Bansal, and R. Matthew Walter (2016). Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In: *30th AAAI Conference on Artificial Intelligence, AAAI 2016*. AAAI press, pp. 2772–2778.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., pp. 3111–3119.

Mills, Gregory J. (2007). Semantic co-ordination in dialogue: the role of direct interaction. PhD thesis. Queen Mary University of London.

— (2014). Dialogue in joint activity: Complementarity, convergence and conventionalization. In: *New Ideas in Psychology* 32, pp. 158–173.

Min, So Yeon, Devendra Singh Chaplot, Pradeep Kumar Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov (2022). FILM: Following Instructions in Language with Modular Methods. In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Misra, Dipendra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi (2018). Mapping Instructions to Actions in 3D Environments with Visual Goal Prediction. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Brussels, Belgium: Association for Computational Linguistics, pp. 2667–2678.

Misra, Dipendra, John Langford, and Yoav Artzi (2017). Mapping Instructions and Visual Observations to Actions with Reinforcement Learning. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1004–1015.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. In: *Deep Learning Workshop at NIPS 2013*.

Montague, R. (1970). Universal Grammar. In: *Theoria* 36, pp. 373–398.

Nagaraja, Varun K., Vlad I. Morariu, and Larry S. Davis (2016). Modeling Context Between Objects for Referring Expression Understanding. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 792–807.

Narayan-Chen, Anjali, Prashant Jayannavar, and Julia Hockenmaier (2019). Collaborative Dialogue in Minecraft. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 5405–5415.

Nesset, Birthe, David A. Robb, José Lopes, and Helen Hastie (2021). Transparency in HRI: Trust and Decision Making in the Face of Robot Errors. In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '21 Companion. Boulder, CO, USA: Association for Computing Machinery, pp. 313–317.

Nguyen, Khanh X, Yonatan Bisk, and Hal Daumé Iii (2022). A Framework for Learning to Request Rich and Contextually Useful Information from Humans. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 16553–16568.

Novikova, Jekaterina, Ondřej Dušek, and Verena Rieser (2017). The E2E Dataset: New Challenges For End-to-End Generation. In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Ed. by Kristiina Jokinen, Manfred Stede, David DeVault, and Annie Louis. Saarbrücken, Germany: Association for Computational Linguistics, pp. 201–206.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh,

Chelsea Finn, and Sergey Levine (2024). Octo: An Open-Source Generalist Robot Policy. In: *Proceedings of Robotics: Science and Systems*. Delft, Netherlands.

OpenAI (2024). GPT-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: 2024-06-12.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022). Training language models to follow instructions with human feedback. In: *Advances in neural information processing systems* 35, pp. 27730–27744.

Owen, Daniel L (1978). The use of influence diagrams in structuring complex decision problems. In: *Readings on the principles and applications of decision analysis* 2, pp. 763–771.

Padmakumar, Aishwarya, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur (2022). TEACh: Task-Driven Embodied Agents That Chat. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.2, pp. 2017–2025.

Pantazopoulos, Georgios, Alessandro Suglia, Oliver Lemon, and Arash Eshghi (2024). Lost in Space: Probing Fine-grained Spatial Understanding in Vision and Language Resamplers. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, pp. 540–549.

Parekh, Amit, Nikolas Vitsakis, Alessandro Suglia, and Ioannis Konstas (2024). Investigating the Role of Instruction Variety and Task Difficulty in Robotic Manipulation Tasks. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 19389–19424.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). GloVe: Global Vectors for Word Representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543.

Pezzelle, Sandro (2023). Dealing with Semantic Underspecification in Multimodal NLP. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 12098–12112.

Piantadosi, Steven T., Harry Tily, and Edward Gibson (2012). The communicative function of ambiguity in language. In: *Cognition* 122.3, pp. 280–291.

Platonov, Georgiy, Lenhart Schubert, Benjamin Kane, and Aaron Gindi (2020). A Spoken Dialogue System for Spatial Question Answering in a Physical Blocks World. In: *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Olivier Pietquin, Smaranda Muresan, Vivian Chen, Casey Kennington, David Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt, and Stefan Ultes. 1st virtual meeting: Association for Computational Linguistics, pp. 128–131.

Purver, Matthew (2004). The Theory and Use of Clarification Requests in Dialogue. PhD thesis. University of London.

— (2006). CLARIE: Handling Clarification Requests in a Dialogue System. In: *Research on Language and Computation* 4.2-3, pp. 259–288.

Purver, Matthew and Jonathan Ginzburg (2004). Clarifying Noun Phrase Semantics. In: *Journal of Semantics* 21.3, pp. 283–339.

Purver, Matthew, Jonathan Ginzburg, and Patrick G. T. Healey (2003). On the Means for Clarification in Dialogue. In: *Current and New Directions in Discourse & Dialogue*. Ed. by R. Smith and J. van Kuppevelt. Kluwer Academic Publishers, pp. 235–255.

Purver, Matthew, Julian Hough, and Christine Howes (2018). Computational Models of Miscommunication Phenomena. In: *Topics in Cognitive Science (topiCS)*. Ed. by Patrick G. T. Healey, Jan de Ruiter, and Gregory J. Mills. Vol. 10.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). Learning Transferable Visual Models From Natural Language Supervision.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). Improving Language Understanding by Generative Pre-Training. In: *Technical report, OpenAI*, pp. 1–10.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). Language Models are Unsupervised Multitask Learners.

Rafailov, Rafael, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn (2024). Direct Preference Optimization: Your Language Model is Secretly a Reward Model.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. In: *Journal of Machine Learning Research* 21.140, pp. 1–67.

Raffin, Antonin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann (2019). Stable Baselines3. `https://github.com/DLR-RM/stable-baselines3`.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, pp. 91–99.

Rieser, Verena and Oliver Lemon (2006). Using Machine Learning to Explore Human Multimodal Clarification Strategies. In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, pp. 659–666.

Rieser, Verena and Johanna Moore (2005). Implications for Generating Clarification Requests in Task-Oriented Dialogues. In: *Proceedings of the 43rd Annual Meeting of the ACL*. Association for Computational Linguistics. Ann Arbor, pp. 239–246.

Robb, David A., Francisco J. Chiyah Garcia, Atanas Laskov, Xingkun Liu, Pedro Patron, and Helen Hastie (2018). Keep Me in the Loop: Increasing Operator Situation Awareness through a Conversational Multimodal Interface. In: *Proceedings of the 20th ACM International Conference on Multimodal Interaction*. ICMI'18. Boulder, Colorado, USA: ACM, pp. 384–392.

Rodríguez, Kepa and David Schlangen (2004). Form, Intonation and Function of Clarification Requests in German Task-Oriented Spoken Dialogues. In: *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*. Barcelona, Spain.

Roman Roman, Homero, Yonatan Bisk, Jesse Thomason, Asli Celikyilmaz, and Jianfeng Gao (2020). RMM: A Recursive Mental Model for Dialogue Navigation. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 1732–1745.

Rosenblatt, Frank (1958). The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological review* 65(6), pp. 386–408.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations*. Ed. by D. E. Rumelhart and J. L. McClelland. Bradford Books/MIT Press, Cambridge, MA., pp. 318–362.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1988). Learning representations by back-propagating errors. In: *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, pp. 696–699.

Salemi, Alireza, Sheshera Mysore, Michael Bendersky, and Hamed Zamani (2024). LaMP: When Large Language Models Meet Personalization. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 7370–7392.

Salin, Emmanuelle, Badreddine Farah, Stéphane Ayache, and Benoit Favre (2022). Are Vision-Language Transformers Learning Multimodal Representations? A Probing Perspective. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.10, pp. 11248–11257.

San-Segundo, Ruben, Juan M. Montero, J. Ferreiros, R. Córdoba, and José M. Pardo (2001). Designing Confirmation Mechanisms and Error Recover Techniques in a Railway Information System for Spanish. In: *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics. Aalborg, Denmark, pp. 136–139.

Saravanos, Antonios, Stavros Zervoudakis, Dongnanzi Zheng, Neil Stott, Bohdan Hawryluk, and Donatella Delfino (2021). The Hidden Cost of Using Amazon Mechanical Turk for Research. In: *Lecture Notes in Computer Science*. Springer International Publishing, pp. 147–164.

Savva, Manolis, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra (2019). Habitat: A Platform for Embodied AI Research. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346.

Schatzmann, Jost, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young (2007). Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In: *Human Language Technologies 2007: The Conference of the North American Chap-*

*ter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Ed. by Candace Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai. Rochester, New York: Association for Computational Linguistics, pp. 149–152.

Schegloff, E.A. (1992). Repair after next turn: The last structurally provided defense of intersubjectivity in conversation. In: *American Journal of Sociology*, pp. 1295–1345.

Schegloff, E.A., Gail Jefferson, and Harvey Sacks (1977). The preference for self-correction in the organization of repair in conversation. In: *Language* 53.2, pp. 361–382.

Schlangen, David (2004). Causes and Strategies for Requesting Clarification in Dialogue. In: *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, pp. 136–143.

— (2023). On General Language Understanding. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 8818–8825.

Schlangen, David, Tim Diekmann, Nikolai Ilinykh, and Sina Zarrieß (2018). slurk–a lightweight interaction server for dialogue experiments and data collection. In: *Proceedings of the 22nd Workshop on the Semantics and Pragmatics of Dialogue (AixDial/semdial 2018)*.

Schlangen, David, Sina Zarrieß, and Casey Kennington (2016). Resolving References to Objects in Photographs using the Words-As-Classifiers Model. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, pp. 1213–1223.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). Proximal Policy Optimization Algorithms.

Searle, J.R. (1980). Minds, brains, and programs. In: *Behavioral and brain sciences* 3.03, pp. 417–424.

Seitzer, Maximilian, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello (2023). Bridging the Gap to Real-World Object-Centric Learning. In: *Proceedings of the Eleventh International Conference on Learning Representations*.

Sermanet, Pierre, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. 2nd International Conference on Learning Representations, ICLR 2014. English (US). In: *Proceedings of the 2nd International Conference on Learning Representations*. ICLR 2014. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Thr joruough 16-04-2014.

Shalyminov, Igor, Arash Eshghi, and Oliver Lemon (2017). Challenging Neural Dialogue Models with Natural Data: Memory Networks Fail on Incremental Phenomena. In: *Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017 - SaarDial)*. Barcelona.

Shekhar, Ravi, Tim Baumgärtner, Aashish Venkatesh, Elia Bruni, Raffaella Bernardi, and Raquel Fernandez (2018). Ask No More: Deciding when to guess in referential visual dialogue. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1218–1233.

Shi, Lei, Kai Shuang, Shijie Geng, Peng Su, Zhengkai Jiang, Peng Gao, Zuohui Fu, Gerard de Melo, and Sen Su (2020). Contrastive Visual-Linguistic Pretraining.

Shi, Zhengxiang, Yue Feng, and Aldo Lipani (2022). Learning to Execute Actions or Ask Clarification Questions. In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan

Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, pp. 2060–2070.

Shridhar, Mohit, Lucas Manuelli, and Dieter Fox (2022). CLIPort: What and Where Pathways for Robotic Manipulation. In: *Proceedings of the 5th Conference on Robot Learning*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, pp. 894–906.

Shridhar, Mohit, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox (2020). ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10737–10746.

Stalnaker, Robert (1978). Assertion. In: *Syntax and Semantics 9: Pragmatics*. Ed. by P. Cole. New York: Academic Press, pp. 315–32.

— (2002). Common Ground. In: *Linguistics and Philosophy* 25.5–6, pp. 701–721.

Steels, Luc (2008). The Symbol Grounding Problem Has Been Solved. So What's Next? English. In: *Symbols and Embodiment: Debates on Meaning and Cognition*. Ed. by M. Vega. Symbols and Embodiment: Debates on Meaning and Cognition. M. Vega. United Kingdom: Oxford University Press.

Stiennon, Nisan, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano (2020). Learning to summarize from human feedback. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc.

Su, Weijie, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai (2020). VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In: *International Conference on Learning Representations*.

Suglia, Alessandro, Ioannis Konstas, and Oliver Lemon (2024). Visually Grounded Language Learning: a review of language games, datasets, tasks, and models. In: *Journal of Artificial Intelligence Research* 79, pp. 173–239.

Suglia, Alessandro, Ioannis Konstas, Andrea Vanzo, Emanuele Bastianelli, Desmond Elliott, Stella Frank, and Oliver Lemon (2020). CompGuessWhat?!: A Multi-task Evaluation Framework for Grounded Language Learning. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 7625–7641.

Suhr, Alane and Yoav Artzi (2018). Situated Mapping of Sequential Instructions to Actions with Single-step Reward Observation. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, pp. 2072–2082.

Suhr, Alane, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi (2019). Executing Instructions in Situated Collaborative Interactions. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 2119–2130.

Sutton, Richard S. and Andrew G. Barto (2018). Reinforcement Learning: An Introduction. 2nd Ed. MIT Press.

Szegedy, Christian, Alexander Toshev, and Dumitru Erhan (2013). Deep neural networks for object detection. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., pp. 2553–2561.

Tan, Hao and Mohit Bansal (2018). Source-target inference models for spatial instruction understanding. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press.

— (2019). LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 5100–5111.

Tellex, Stefanie, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy (2014). Asking for Help Using Inverse Semantics. In: *Robotics: Science and Systems X*. RSS2014. Robotics: Science and Systems Foundation.

Tellex, Stefanie, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy (2011). Understanding natural language commands for robotic navigation and mobile manipulation. In: *Twenty-Fifth AAAI Conference on Artificial Intelligence*. Vol. 2. AAAI 2011, pp. 1507–1514.

Thomason, Jesse, Daniel Gordon, and Yonatan Bisk (2019). Shifting the Baseline: Single Modality Performance on Visual Navigation & QA. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1977–1983.

Thomason, Jesse, Michael Murray, Maya Cakmak, and Luke Zettlemoyer (2020). Vision-and-Dialog Navigation. In: *Proceedings of the Conference on Robot Learning*. Ed. by Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, pp. 394–406.

Thomason, Jesse, Shiqi Zhang, Raymond Mooney, and Peter Stone (2015). Learning to interpret natural language commands through human-robot dialog. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, pp. 1923–1929.

Tomasello, Michael (1995). Joint attention as social cognition. In: *Joint attention : Its origins and role in development*, pp. 103–130.

Tomasello, Michael, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll (2005). Understanding and sharing intentions: The origins of cultural cognition. In: *Behavioral and Brain Sciences* 28.5, pp. 675–691.

Torruella, Joan and Ramon Capsada (2013). Lexical Statistics and Tipological Structures: A Measure of Lexical Richness. In: *Procedia - Social and Behavioral Sciences* 95, pp. 447–454.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample (2023a). LLaMA: Open and Efficient Foundation Language Models.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan

Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom (2023b). Llama 2: Open Foundation and Fine-Tuned Chat Models.

Udandarao, Vishaal, Abhishek Maiti, Deepak Srivatsav, Suryatej Reddy Vyalla, Yifang Yin, and Rajiv Ratn Shah (2020). COBRA: Contrastive Bi-Modal Representation Algorithm.

Van der Maaten, Laurens and Geoffrey Hinton (2008). Visualizing data using t-SNE. In: *Journal of machine learning research* 9.11.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., pp. 6000–6010.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2019). SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Tal Linzen, Grzegorz Chrupała, and Afra Alishahi. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355.

Wang, Liyuan, Xingxing Zhang, Hang Su, and Jun Zhu (2024). A Comprehensive Survey of Continual Learning: Theory, Method and Application. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.8, pp. 5362–5383.

Wang, Peng, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang (2022). Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: *International Conference on Machine Learning*. PMLR, pp. 23318–23340.

Watkins, Christopher J. C. H. and Peter Dayan (1992). Q-learning. In: *Machine Learning* 8.3–4, pp. 279–292.

Wei, Jason, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le (2022). Finetuned Language Models Are Zero-Shot Learners.

Willemsen, Bram, Livia Qian, and Gabriel Skantze (2023). Resolving References in Visually-Grounded Dialogue via Text Generation. In: *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani. Prague, Czechia: Association for Computational Linguistics, pp. 457–469.

Willemsen, Bram and Gabriel Skantze (2024). Referring Expression Generation in Visually Grounded Dialogue with Discourse-aware Comprehension Guiding. In: *Proceedings of the 17th International Natural Language Generation Conference*. Ed. by Saad Mahamood, Nguyen Le Minh, and Daphne Ippolito. Tokyo, Japan: Association for Computational Linguistics, pp. 453–469.

Williams, Jason D., Kavosh Asadi, and Geoffrey Zweig (2017). Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, pp. 665–677.

Wilson, Bruce W, Shivaanee Eswaran, Omar Riyaz, Javier Chiyah-Garcia, and Matthew Peter Aylett (2024). Follow the Yellow or Red Brick Road? Investigating the Impact of Narratives in a Guided Navigation Task. In: *Proceedings of the 12th International*

*Conference on Human-Agent Interaction*. HAI '24. Swansea, United Kingdom: Association for Computing Machinery, pp. 411–413.

Winograd, Terry (1972). Understanding natural language. In: *Cognitive Psychology* 3.1, pp. 1–191.

Wright, Julia L., Jessie Y. C. Chen, and Shan G. Lakhmani (2020). Agent Transparency and Reliability in Human–Robot Interaction: The Influence on User Confidence and Perceived Reliability. In: *IEEE Transactions on Human-Machine Systems* 50.3, pp. 254–263.

Wu, Chien-Sheng, Steven C.H. Hoi, Richard Socher, and Caiming Xiong (2020a). TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 917–929.

Wu, Qi, Peng Wang, Xin Wang, Xiaodong He, and Wenwu Zhu (2022). Referring Expression Comprehension. In: *Visual Question Answering: From Theory to Application*. Singapore: Springer Nature Singapore, pp. 219–230.

Wu, Yi, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian (2018). Building Generalizable Agents with a Realistic and Rich 3D Environment. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Wu, Yunhan, Daniel Rough, Anna Bleakley, Justin Edwards, Orla Cooney, Philip R. Doyle, Leigh Clark, and Benjamin R. Cowan (2020b). See What I'm Saying? Comparing Intelligent Personal Assistant Use for Native and Non-Native Language Speakers. In: *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '20. Oldenburg, Germany: Association for Computing Machinery.

Wu, Yuxin, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick (2019). Detectron2. `https://github.com/facebookresearch/detectron2`.

Yan, Claudia, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi (2018). CHALET: Cornell House Agent Learning Environment.

Yu, Licheng, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg (2016a). Modeling Context in Referring Expressions. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 69–85.

Yu, Yanchao, Arash Eshghi, and Oliver Lemon (2016b). Training an adaptive dialogue policy for interactive learning of visually grounded word meanings. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Raquel Fernandez, Wolfgang Minker, Giuseppe Carenini, Ryuichiro Higashinaka, Ron Artstein, and Alesia Gainer. Los Angeles: Association for Computational Linguistics, pp. 339–349.

Zhai, Xiaohua, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer (2023). Sigmoid Loss for Language Image Pre-Training. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986.

Zhang, Hanwang, Yulei Niu, and Shih-Fu Chang (2018). Grounding Referring Expressions in Images by Variational Context. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4158–4166.

Zhang, Yizhe, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan (2020). DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Ed. by Asli Celikyilmaz and Tsung-Hsien Wen. Online: Association for Computational Linguistics, pp. 270–278.

Zhu, Yaxin and Hamed Zamani (2024). ICXML: An In-Context Learning Framework for Zero-Shot Extreme Multi-Label Classification. In: *Findings of the Association for Computational Linguistics: NAACL 2024*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, pp. 2086–2098.

Zhu, Yi, Yue Weng, Fengda Zhu, Xiaodan Liang, Qixiang Ye, Yutong Lu, and Jianbin Jiao (2021). Self-Motivated Communication Agent for Real-World Vision-Dialog Navigation. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1574–1583.

Ziegler, Daniel M., Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving (2019). Fine-Tuning Language Models from Human Preferences.