Situated Grounding and Understanding of Structured Low-resource Expert Data

Miltiadis Marios Katsakioris

Submitted for the degree of
Doctor of Philosophy

Heriot-Watt University

Department of Mathematics,
School of Mathematical and Computer Sciences.

August, 2022

Abstract

Conversational agents are becoming more widespread, varying from social to goal-oriented to multi-modal dialogue systems. However, for systems with both visual and spatial requirements, such as situated robot planning, developing accurate goal-oriented dialogue systems can be extremely challenging, especially in dynamic environments, such as underwater or first responders. Furthermore, training data-driven algorithms in these domains is challenging due to the esoteric nature of the interaction, which requires expert input. We derive solutions for creating a collaborative multi-modal conversational agent for setting high-level mission goals. We experiment with state-of-the-art deep learning models and techniques and create a new data-driven method (MAPERT) that is capable of processing language instructions by grounding the necessary elements using various types of input data (vision from a map, text and other metadata). The results show that, depending on the task, the accuracy of data-driven systems can vary dramatically depending on the type of metadata and the attention mechanisms that are used. Finally, we are dealing with low-resource expert data and this inspired the use of the Continual Learning and Human In The Loop methodology with encouraging results.

To my parents.

Acknowledgements

First of all, I want to express my gratitude to my supervisor, Helen Hastie, for her motivation and assistance throughout my PhD and for providing invaluable feedback. I am also very thankful to my second supervisor Ioannis Konstas, for being there to offer his technical expertise, a new insight and new ideas. I would also like to thank my industry supervisors Pedro Patron and Pierre-Yves Mignotte for the hours that we spent together brainstorming and the fruitful conversations.

I am very grateful to my funders, Seebyte, Datalab and MASTS-S without whom I could not have the resources to perform my research.

I am very grateful to the iLab members and alumni, Dr Igor Shalyminov, Dr Shubham Agarwal, Dr Christian Dondrup, Dr Ioannis Papaioannou, Dr Frank Broz, Dr Boris Mocialov, Dr David Howcroft, Dr Xingkun Liu, Dr Ondřej Dušek, Dr Katrin Lohan, Dr José David Águas Lopes, Dr Amanda Cercas Curry, Dr Jose Part, Dr Andrea Vanzo, Dr Angus Addlesee, Dr Karin Sevegnani, Dr Alessandro Suglia, Dr Xinnuo Xu, Dr Yanchao Yu, Prof Verena Rieser and Prof Oliver Lemon. I'd like to thank Derek Davis in particular for always being there for a nice chat, even during the hardest of times.

I am fortunate to have friends who supported me in their own way, as well as flatmates who did the same, Cero for spending hours with me at the lab, Beba for her life lessons, Tiko for cooking with me, my whole family in Greece and the Zavitsanos family for their inspiring push to success. Finally, I am endlessly grateful to my mother, as well as Melirrytos for their unconditional love and support throughout this life-changing journey.

HERIOT WATT UNIVERSITY

Research Thesis Submission

Please note this form should be bound into the submitted thesis.

Name*:*	*Miltiadis Marios Katsakioris*		
School:	Mathematical & Computer Sciences		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought:	*PHD*

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	06/08/2022

Submission

Submitted By *(name in capitals):*	MILTIADIS MARIOS KATSAKIORIS
Signature of Individual Submitting:	
Date Submitted:	06/08/2022

For Completion in the Student Service Centre (SSC)

Limited Access	Requested	Yes		No		Approved	Yes		No	
E-thesis Submitted **(*mandatory for final theses*)**									
Received in the SSC by *(name in capitals):*					Date:				

Inclusion of Published Works

Please note you are only required to complete the Inclusion of Published Works Form if your thesis contains published works under Regulation 6 (9.1.2)

Declaration

This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:

Citation details	Miltiadis Marios Katsakioris and Helen Hastie and Ioannis Konstas and Atanas Laskov, Corpus of Multi-modal Interaction for Collaborative Planning, In Proceedings of the SpLU-RoboNLP Workshop in conjunction with the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, USA (2019)
Author 1	
Author 2	
Signature:	
Date:	06/08/2022

Citation details	Miltiadis Marios Katsakioris and Ioannis Konstas and Pierre Yves Mignotte and Helen Hastie, ROSMI: A multi-modal Corpus for Map-based Instruction-Giving, In Proceedings of the International Conference on Multimodal Interaction (ICMI), Association for Computing Machinery (ACM), New York, NY, USA (2020)
Author 1	
Author 2	
Signature:	
Date:	

Citation details	Miltiadis Marios Katsakioris and Ioannis Konstas and Pierre Yves Mignotte and Helen Hastie, Learning to Read Maps: Understanding Natural Language Instructions from Unseen Maps, In Proceedings of the Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP), ACM, Bangkok, Thailand (2021)
Author 1	
Author 2	
Signature:	
Date:	06/08/2022

Citation details	Katsakioris, M. M., Zhou, Y., and Masato, D. (2022). Entity linking in tabular data needs the right attention.
Author 1	
Author 2	
Signature:	
Date:	06/08/2022

Please included additional citations as required.

Contents

List of Tables

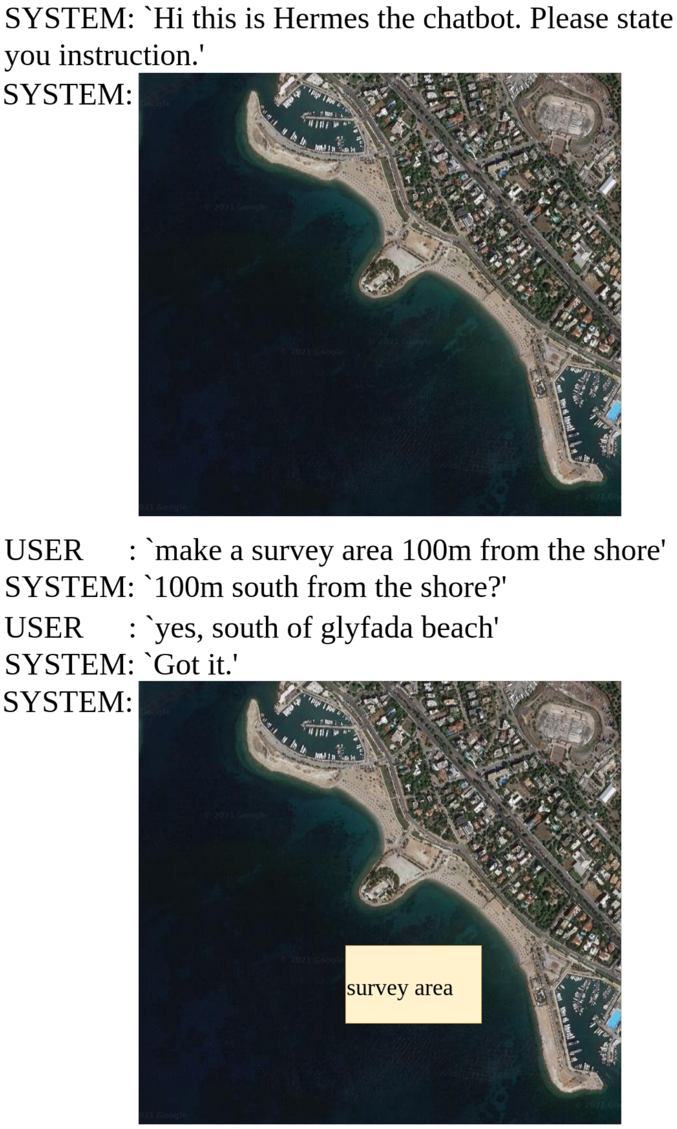
List of Figures

Chapter 1

Introduction

Collaboration between human experts and remote mobile robots is highly challenging, particularly in fast-moving dynamic environments, such as high-stakes first responder scenarios or offshore energy platform inspection (Hastie et al., 2018a). The goal of this thesis is to explore methods for collaborative multi-modal robot planning in the form of a conversational agent (CA), using both visual and natural language interaction (see Figure 1.1). In this work, we focus on the emergency response domain with unmanned systems (henceforth referred to as UXV). Experts in this domain typically create a plan for vehicles using a visual interface on dedicated hardware on-shore, days before the mission. This planning process is complicated and requires expert knowledge. We present a 'planning assistant' that is able to encapsulate this expert knowledge and guide the user through the planning process using natural language multi-modal interaction. To deal with processing natural language instructions and grounding information with a variety of input data (vision from a map, text and other metadata), we investigate various state-of-the-art (SOTA) deep learning models and methodologies, so as to develop a new data-driven method (MAPERT). This will allow for more efficient planning and reduce operator training time. Due to the intuitive nature of language, it will also allow for anywhere access to planning and, importantly, in-situ replanning in fast-moving dynamic scenarios.

Controlling remote mobile robots presents a new opportunity for human-robot interaction but with new challenges. CAs are becoming more widespread, varying from social (Li et al., 2016a) and goal-oriented (Wen et al., 2017) to multi-modal dialogue systems, such as for the Visual Dialogue Challenge (Das et al., 2017), where an AI agent must hold a dialogue with a human in natural language about a given visual content. In systems with both visual and spatial requirements, such as situated robot planning (Misra et al., 2018), developing accurate goal-oriented dialogue systems can be extremely challenging. This is further compounded in the context of remote robots, especially ones we cannot see, such as Autonomous Underwater Vehicles (AUV).



SYSTEM: `Hi this is Hermes the chatbot. Please state you instruction.'
SYSTEM:
USER : `make a survey area 100m from the shore'
SYSTEM: `100m south from the shore?'
USER : `yes, south of glyfada beach'
SYSTEM: `Got it.'
SYSTEM:

Figure 1.1: Dialogue excerpt from the system presented in Chapter 3 and the corresponding images.

In situated collaborative planning, each user must be able to comprehend the environment at hand and referring expressions need to be carefully selected and verified, to avoid perceiving the environment in different or contradictory ways, especially if the shared environment is ambiguous (Fang et al., 2013). Therefore, to achieve collaboration between an agent and a human on a specific context, an agent needs to be able to understand situated instructions and ground them in the environment as it is being perceived by the human expert, so as to make decisions and plan together successfully. Hereinafter, challenges on the expert domain of emergency response with UXV are discussed.

1.1 Motivation

Robots and autonomous systems (RAS) are becoming commonplace for tasks where it is dangerous for humans to go, such as deep underwater. These systems may be deployed for inspection or emergency response in off-shore energy installations, defence or search and rescue (Kwon and Yi, 2012; Nagatani et al., 2013; Shukla and Karki, 2016; Wong et al., 2017). Imminently, teams of remote robots will be required to inspect remote off-shore platforms or act in first responder scenarios (Hastie et al., 2018b). Rich structured data, such as plans of buildings, detailed maps or charts (e.g. Electronic Navigational Charts (ENC) or OpenStreetMaps (OSM)) are the key for successful deployment of RAS in these scenarios.

OSM are a crowdsourced collection of editable maps of the world. The data is available under the Open Data Commons Open Database Licence (ODbL) and has been broadly used for prior research studying reference resolution, landmark salience or geographical dialogue systems (Götze and Boye, 2016; Hentschel and Wagner, 2010; Haklay and Weber, 2008). It is a collection of publicly available, layman-friendly geodata (see Figure 1.2). As well as OSM, in the work described here we also use ENC, which are the official databases created by the National Hydrographic Office for use with an Electronic Chart Display and Information System (ECDIS) and must conform to certain standards before being certified as an ENC (see Figure 1.2). ENCs are considerably denser than the OSM and typically readable only to experts, such as maritime operators.

In the work described here, such structured geodata act as the "world" of interaction in the domain of inspection and emergency response. However, geodata is not typically easily digestible to humans, which can be particularly problematic in fast-moving collaborative scenarios such as emergency response, where there are a variety of stakeholders and players. Thus, the objective is to have CAs collabo-

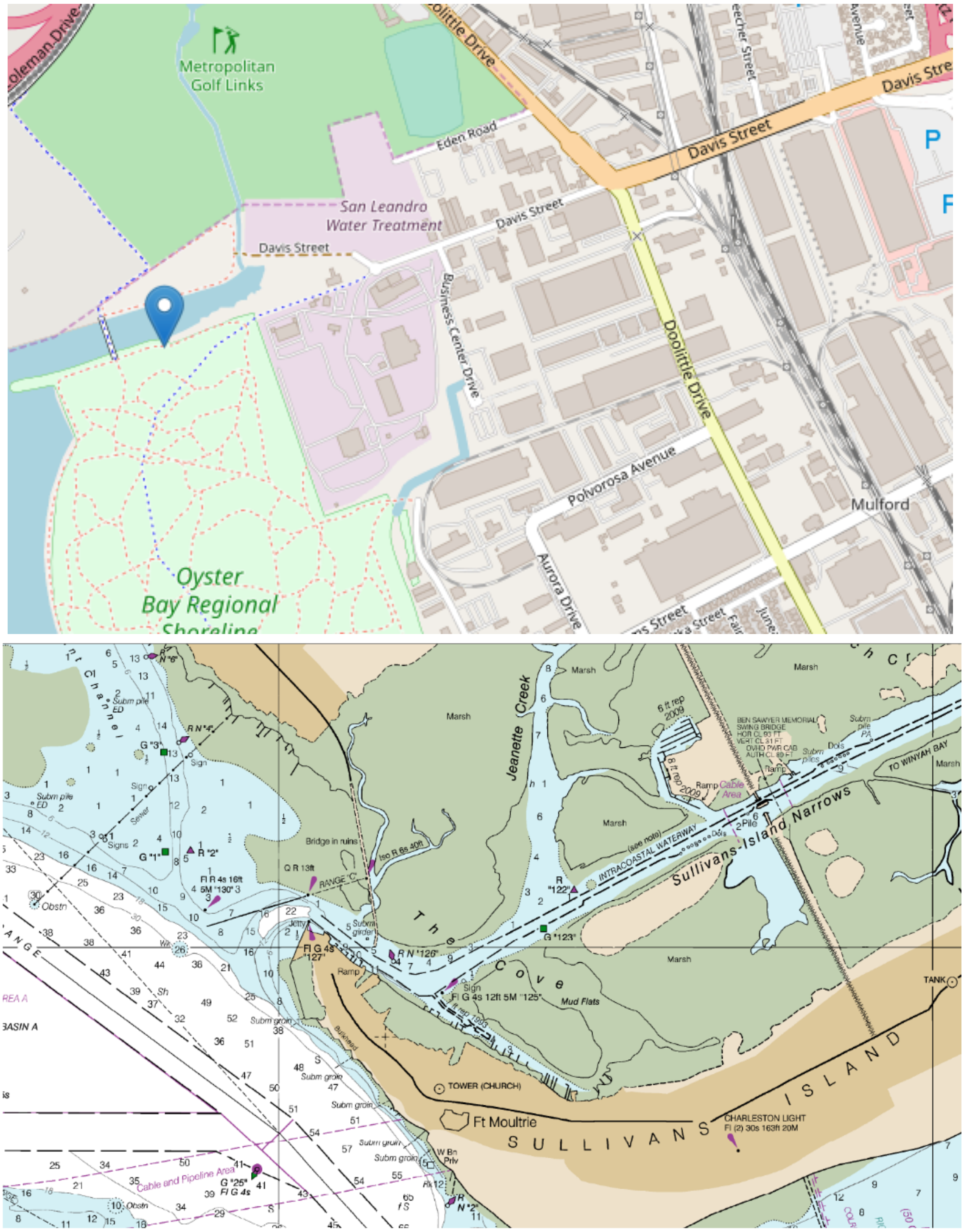


Figure 1.2: OpenStreetMap (OSM) map above and an Electronic Navigational Chart (ENC) chart below (Sources: `https://www.openstreetmap.org/` and `https://charts.noaa.gov/InteractiveCatalog/nrnc.shtml` respectively).

rate with human experts and act as "mediators" between these remote robots and humans. This will allow faster, improved action execution and self-organisation of teams of robots. Commonly, these robots have been remotely operated by humans and this new dynamic requires a shift from user-manipulation of a tool to a collaborative relationship between human and machine (Phillips et al., 2011).

Despite the recent advances of CAs (Chai et al., 2016; Serban et al., 2016a), there have been few CAs performing such tasks in the real world (e.g. security agents, healthcare assistants, tutors etc.), with most of them offering pure unimodal social interaction, such as Alexa (Lopatovska et al., 2019) or DeepProbe (Yin et al., 2017), which are able to execute limited actions in the real world (e.g. switching on a light). There has been prior work on multi-modal interfaces, whereby a human can interact with voice, images or text and have a CA respond through multi-modal output (e.g. vision and voice/text), thus situating interaction in a common ground allowing faster and less error-prone communication (Hastie et al., 2017a; Oviatt et al., 2004). However, real world interaction comes with high risk, especially on high stake domains, such as emergency response with UXV, where expert knowledge is required and prediction errors can have catastrophic consequences.

UXVs represent a wide variety of unmanned vehicles: (1) Unmanned Aerial Vehicles (UAV); (2) Unmanned Surface Vehicles (USV); (3) Unmanned Ground Vehicles (UGV); (4) Unmanned Underwater Vehicles (UUV), which are divided in Remotely Operated Underwater Vehicles (ROUV) and Autonomous Underwater Vehicles (AUV) (see Figure 1.3). In this thesis, our conversational agent will work with planning software called SeeTrack provided by our industrial partner SeeByte, who mainly focus on maritime autonomy, such as AUV and USV. Hastie et al. (2017b) propose MIRIAM, a prototype spoken dialogue system, with a command and control (C2) interface that uses natural language and visual imagery to interact with several remote UXV. MIRIAM can provide operators with alerts, updates and explanations of events in a mixed-initiative conversation along with processing and acting on them. However, it is not built to understand complex structured data, such as ENC or OSM, and cannot work in unison with human experts on in-situ replanning in fast-moving dynamic environments.

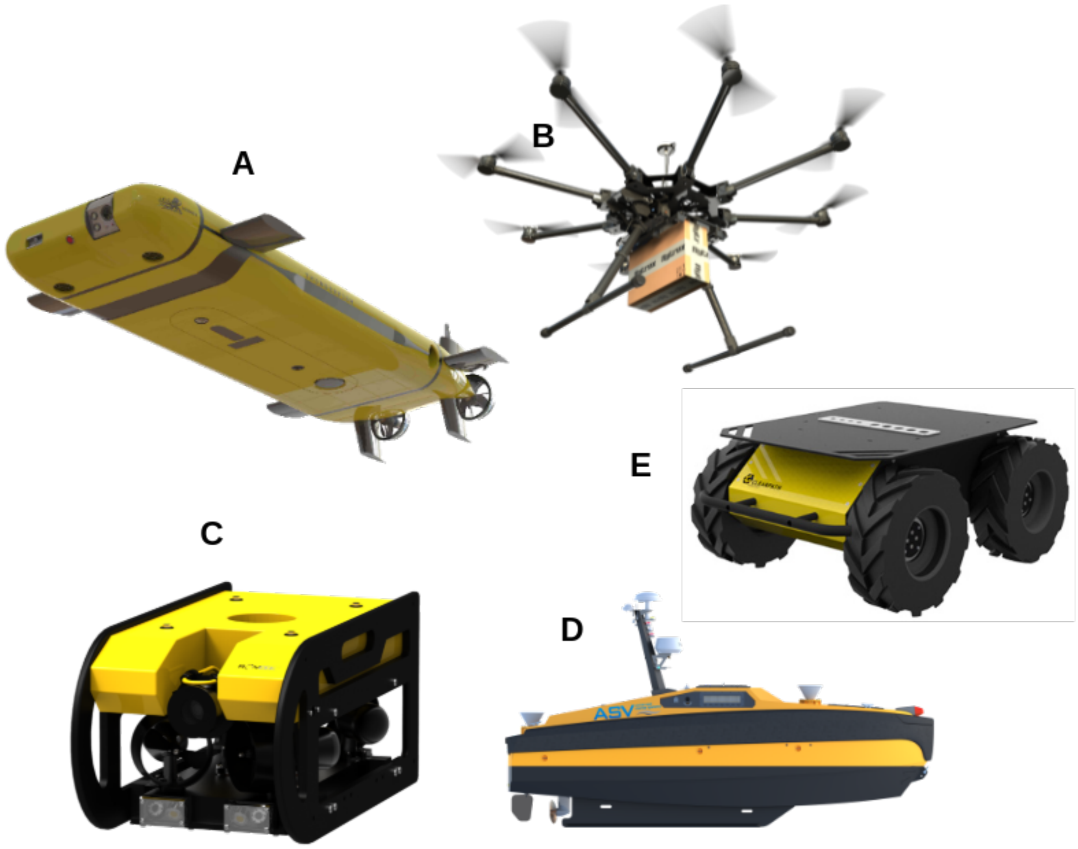


Figure 1.3: Examples of UXV. **A:** Autonomous Underwater Vehicle (AUV); **B:** Unmanned Aerial Vehicle (UAV); **C:** Remotely Operated Underwater Vehicle (ROUV); **D**: Unmanned Surface Vehicle (USV) **E**: Unmanned Ground Vehicle (UGV) (Sources: `https://www.pngegg.com/`, `https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/`).

1.2 Challenges Addressed in this Thesis

After establishing the domain and goal of this thesis, in this section certain challenges, that arise due to the limited availability of information on low resource domains, are discussed below:

1. **Planning and executing in human-robot teams (Chapter 3):** In the real world and more specifically in high risk scenarios, generating safe and efficient plans for remote robots and being able to communicate them through natural language and visualisation, is a complex task that has not been achieved yet (Hastie et al., 2019). It requires SOTA situation awareness comparable or better than that of a human expert. In an effort to understand how a mediator CA would work, a thorough investigation on the way human experts would collaboratively make plans via a CA is explored in Chapter 3.

2. **Data scarcity (Chapters 3 and 4):** Being able to learn from limited amounts of data and generalise to unseen environments, is a consistent issue with low resource expert domains. It is challenging to predict how an algorithm would operate in these dynamic domains and what language operators would use. This effectively means that not only do we not have enough information in order to build Natural Language Processing (NLP) algorithms for these domains, but we also cannot outsource data collections to collect annotated data in order to train data-driven systems. To overcome this barrier, interaction with real experts needs to be examined, so as to acquire relevant information before attempting to work on an NLP algorithm for understanding situated instructions (Chapters 3 and 4).
3. **Understanding the semantics of structured data (Chapter 5):** Understanding the semantics of a world represented by maps, images or other metadata is crucial because otherwise it would be impossible to infer the world that the natural language instruction is referring to. Essentially reading maps includes the following subtasks: (1) *landmark grounding*: the subtask of grounding any language expression from the instruction to a real world landmark or entity, e.g. *"send husky11 62m to the west direction"*; (2) *spatial relation grounding*: the subtask of understanding spatial relations in the environment, such as the "bearing" or direction from one landmark to an other, e.g. *"send drone to southern end of pond22"*. This is investigated thoroughly in Chapter 5.
4. **Generalisability (Chapter 6):** An expert domain by definition comes with specialised features and characteristics that only a few trained human experts are able to process. This implies that for expert domains, specialised algorithms are needed, compared to non-expert domains that may share solutions with minor adjustments. A generalisable NLP algorithm that can be tested in more than one expert domain can provide important insights on the capabilities of the said algorithm. To investigate this, our instruction understanding algorithm is tested across multiple domains in Chapter 6.
5. **Adaptivity to new challenges (Chapter 7):** It would be overly expensive, if a new solution needed to be produced every time new challenges arose, or

as the requirements of experts change. In an effort to avoid having to build a system from scratch and to save time and resources, Chapter 7 looks into the possibilities of continual learning with human intervention using the Human-In-The-Loop (HITL) learning framework.

1.3 Research Questions

The main research question of this thesis is thus:
" Can a multi-modal agent act as a 'mediator' between expert operators and remote UXV, by understanding situated instructions for inspection and emergency response scenarios?"
Following, this research question is broken down into finer-grained questions and then will be addressed in turn in this thesis.

1. How to address the challenge of data scarcity on low-resource expert domains, such as when working with Autonomous Underwater Vehicles (AUV) (Chapters 3 and 4)?
2. How do we develop an end-to-end neural model that understands natural language instructions referring to entities on maps/charts, i.e. that can read maps (Chapter 5)?
3. Is the end-to-end model, developed in (2), task-specific or does it generalise to new tasks (Chapter 6)?
4. How do we embed the model in (2) into a Human-In-the-loop framework for continual improvement through short interactions with experts (Chapter 7)?

1.4 Contributions

This thesis seeks to develop novel approaches to address the research questions presented in the previous section. Its contributions are depicted on Table 1.1.

	Research Question	**Finding**
RQ1 (Ch.3&4)	How to address the challenge of data scarcity on low-resource expert domains, such as when working with Autonomous Underwater Vehicles (AUV)?	Identified the importance of object referencing on the emergency response domain for successful interaction during mission planning. User feedback showed that multi-modality is key to successful interaction. Output, include three datasets, one from a Wizard of Oz study, one based on maps (ROSMI) and one based on charts (RENCI).
RQ2 (Ch.5)	How do we develop an end-to-end neural model that understands natural language instructions referring to entities on maps/charts, i.e. that can read maps?	Findings include a novel, end-to-end, deep learning model (MAPERT) that enables multi-modal fusion of features, is capable of processing situated language instructions and ground them on different environments, such as OSM maps, using various types of input data, and can do so for maps and charts that it has not seen before.
RQ3 (Ch.6)	Is the end-to-end model, developed in (2), task-specific or does it generalise to new tasks ?	Findings are that the task itself and the input features can greatly affect the performance of the model, regardless of the dataset, but with the core of MAPERT intact and minor modifications we can achieve desirable performance in new tasks or domains (hybrid-MAPERT).
RQ4 (Ch.7)	How do we embed the model in (2) into a Human-In-the-loop framework for continual improvement through short interactions with experts?	The model in (2) can be improved over time using the HITL framework with synthetic annotated data without the intervention of an engineer.

Table 1.1: Research questions and findings across chapters.

1.5 Publications

Part of the work presented here has been published and presented in peer-reviewed conferences and workshops:

1. Miltiadis Marios Katsakioris, Helen Hastie, Ioannis Konstas, Atanas Laskov. Corpus of Multi-modal Interaction for Collaborative Planning. In Proceedings of the SpLU-RoboNLP 2019 Workshop in conjunction with the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, USA (2019).
2. Miltiadis Marios Katsakioris, Ioannis Konstas, Pierre Yves Mignotte, and Helen Hastie. ROSMI: A multi-modal Corpus for Map-based Instruction-Giving. In Proceedings of the 2020 International Conference on Multi-modal Interaction (ICMI'20). Association for Computing Machinery (ACM), New York, NY, USA (2020).
3. Miltiadis Marios Katsakioris, Ioannis Konstas, Pierre Yves Mignotte and Helen Hastie. Learning to Read Maps: Understanding Natural Language Instructions from Unseen Maps. In Proceedings of the Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP). Association for Computing Machinery (ACM), Bangkok, Thailand (2021).
4. Miltiadis Marios Katsakioris, Yiwei Zhou, Daniel Masato. Entity Linking in Tabular Data Needs the Right Attention. [UNDER REVIEW] (2022) (see Appendix C).

1.6 Thesis Overview

The remainder of the thesis is organised as follows:

- **Chapter 2** discusses the background; it introduces the symbol grounding problem and how it relates to situated dialogue. It provides essential terminology of conversational agents, language grounding, referring expressions and continual learning. Previous works using the Wizard of Oz methodology explores the task of GPS prediction and SOTA methods on visual grounding.

Last, but not least, the chapter concludes with an overview of the Human-in-the-loop learning framework as a way of continual learning and similar datasets and tasks on instruction giving and navigation.

- **Chapter 3** describes the two-wizard WoZ study, as part of RQ1, to investigate the way human operators would collaboratively make plans with a CA "mediator" that controls UXV remotely.
- **Chapter 4** introduces a new challenge, Robot Open Street Map Instructions (ROSMI)[1], a task and a rich multi-modal dataset of map and instruction understanding with the goal of advancing reference resolution and creating an NLP algorithm that can translate situated natural language instructions to GPS locations for UXV (RQ2). In addition, it introduces a synthetically generated dataset (RENCI)[2], similar to ROSMI but based on ENC, as a continuation of the study done in Chapter 3. The code and data are freely available.
- **Chapter 5** develops an end-to-end data driven model (MAPERT)[3] for predicting GPS goal locations from a map-based natural language instruction that is able to understand instructions referring to previously unseen maps (RQ2). It presents the experimental methodology, and ablation and an error analysis.
- **Chapter 6** examines the performance of the NLP algorithm developed in Chapter 5, with various input metadata on a new task and shows the results of an ablation study (RQ3). Furthermore, it presents an evaluation on the synthetically generated dataset RENCI described in Chapter 4, together with minor modifications of MAPERT necessary to achieve high performance.
- **Chapter 7** aims to evaluate MAPERT with real human subjects and, coming from the field of Continual and Human In The Loop learning (HITL), to answer whether it improve online by incorporating human interaction into the learning process (RQ4).
- **Chapter 8** summarises the main findings and contributions of this thesis and suggests possible avenues for future work.

[1] `https://github.com/marioskatsak/rosmi-dataset`

[2] `https://github.com/marioskatsak/renci-dataset`

[3] `https://github.com/marioskatsak/mapert`

Chapter 2

Literature Review

This chapter provides an overview of conversational agents and how situated instruction understanding has been explored by the research community. Details are provided of the common datasets and methods used in vision-language grounding and GPS location prediction. In addition, a broad overview of the use of attention mechanisms in vision-language tasks is given and last but not least, Human In The Loop (HITL) learning is introduced and how it is being used by the community to train low resource data.

The aim is to give an introduction to conversational agents and situated dialogue, so as to provide a background to readers who are less familiar with interactive systems in the form of agents. Instruction understanding is explored together with vision and language grounding for GPS prediction, in order to get a wider perspective on the NLP algorithms that can solve spatial challenges in dynamic environments, such as the environment of our domain of AUVs. Focusing on methods, such as transfer learning and HITL learning, so as to alleviate the low resource challenge of expert domains.

Section 2.1 goes into further depth about conversational agents, the symbol grounding problem, and how it relates to situated dialogue. Following that, Section 2.2 delves into visual and language grounding, instruction comprehension, and the challenge of GPS prediction. Then, in Section 2.3, the constraints of expert domains are examined and the Human-in-the-Loop (HITL) Learning Framework and Transfer Learning are discussed. Finally, Section 2.4 summarises the chapter.

2.1 Conversational Agents & Situated Dialogue Systems

Natural language provides a powerful interface for interaction between humans and robots. Hence, an agent that can understand and follow instructions is one of the longstanding challenges of AI, first introduced during the 70s as SHRDLU (Winograd, 1972). Turn-based conversational agents try to comprehend the world and communicate with users in natural language and fall into two categories as described by Jurafsky and Martin (2000):

- **Goal-oriented**, systems that use conversation with users to help complete tasks. Some examples are finding restaurants, answering questions, giving directions or controlling appliances.
- **Chat-oriented**, systems designed for extended conversations, set up to mimic the unstructured conversations mainly for entertainment, "chit-chat", or making goal-oriented agents speak more natural.

Both Goal-oriented and Chat-oriented systems have the ability to converse in a dialogue with a user. A dialogue is a sequence of **turns** usually between a system and a human. Depending on the capabilities and needs of the system, the initiative of the dialogue might be controlled by the user, **user-initiative**, the system, **system-initiative** or both, **mixed-initiative**, which tend to be the most complex systems because of their dynamic nature. As Austin (1962) noted, an utterance in a dialogue is kind of an "action" performed by the speaker. These actions are called speech or dialogue acts. Example dialogue act types categorised in three groups, adapted from the ISO (24617-2:2012) standard for dialogue act annotation, are shown below:

- Generic (conversational acts): wait, ack, affirm, yourwelcome, thankyou, bye, hello, repeat, praise, apology.
- Informative: inform, negate, delete, create, correction.
- Request: request, enqmore.

Since dialogue is a sequence of acts, it is crucial for the participants to establish common ground, through **grounding**. Participants continuously ground each other's utterances, which can be explicit e.g. saying "OK", or by repeating what the other person says (Jurafsky and Martin, 2000). This is different from **visual grounding**, which is the task of locating an object instance referred by the participants in the conversation. Since chat-oriented systems are out of this thesis's scope, in the next subsection, we focus on goal-oriented dialogue systems.

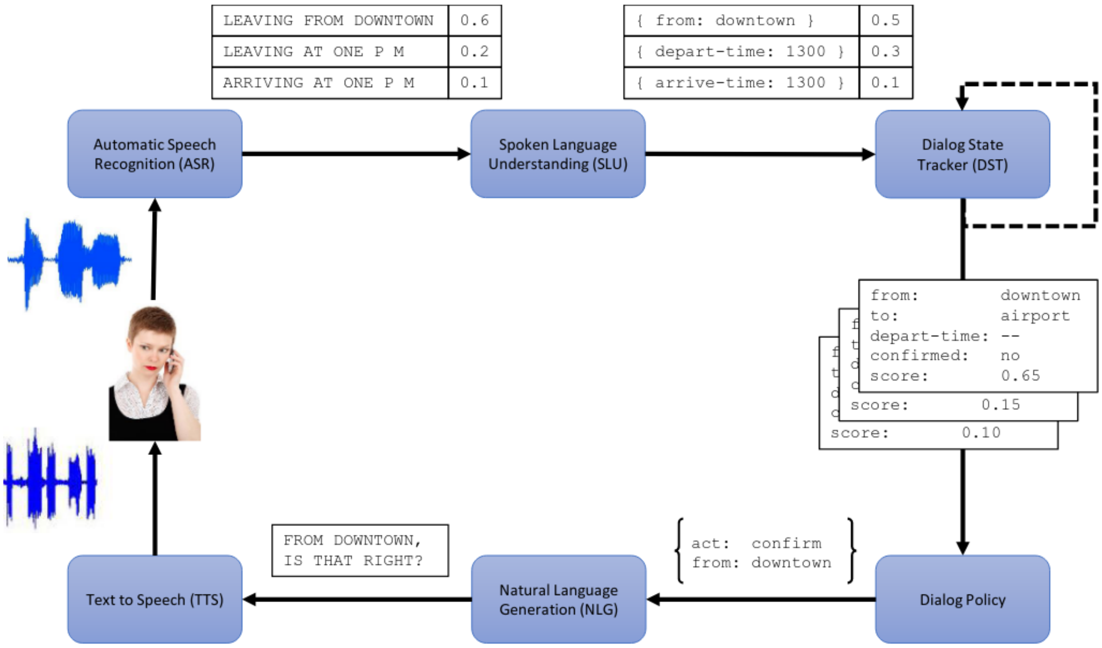

LEAVING FROM DOWNTOWN 0.6
LEAVING AT ONE P M 0.2
ARRIVING AT ONE P M 0.1
{ from: downtown } 0.5
{ depart-time: 1300 } 0.3
{ arrive-time: 1300 } 0.1
Automatic Speech Recognition (ASR)
Spoken Language Understanding (SLU)
Dialog State Tracker (DST)
from: downtown
to: airport
depart-time: --
confirmed: no
score: 0.65
score: 0.15
score: 0.10
act: confirm
from: downtown
FROM DOWNTOWN, IS THAT RIGHT?
Text to Speech (TTS)
Natural Language Generation (NLG)
Dialog Policy

Figure 2.1: Overview of a Dialogue-state architecture for goal-oriented dialogue systems (Williams et al., 2016).

2.1.1 Goal-oriented Dialogue

Broadly there are two lines of work applying machine learning to dialogue control. The first one includes systems for goal-oriented dialogue that are based on the dialogue-state architecture (Levin et al., 2000; Singh et al., 2002; Williams, 2008; Lee et al., 2009; Li et al., 2014). Figure 2.1 shows the six components of this architecture. The first and last component, the Automatic Speech Recognition (ASR) and Text to Speech (TTS) components are out of the scope of this thesis. The remaining components are the following:

The **Natural Language Understanding (NLU)** or Spoken Language Understanding (SLU) is used for extracting slot values from the user's utterance either by machine learning or hand-crafted rules. The next component, the **Dialogue State Tracker (DST)** maintains the current state of the dialogue, which usually includes the user's most recent dialogue act, dialogue history, and the entire set of slot-values the user has expressed so far. Following DST, is the **Dialogue Policy (DP)** an "action" policy that decides what the system should do or say next. In the literature, DST and DP are represented together as the **Dialogue Manager (DM)**. Finally, there is the **Natural Language Generation (NLG)** component, which is responsible for synthesising the utterance of the next turn. In the aforementioned works, the policy depends on the DST to summarise observable dialogue history into state features, which requires design and specialised labelling. The second, more recent line of work, takes a different approach and instead of a dialogue architecture, tries to learn "end-to-end" deep learning models, which map from an observable dialogue history directly to a sequence of output words. The most broad are, either Recurrent Neural Network (RNN) based (Serban et al., 2016b; Shang et al., 2015; Mei et al., 2017; Lowe et al., 2017; Luan et al., 2016) or transformer based (Wolf et al., 2019; Haihong et al., 2019; Gou et al., 2021; Le et al., 2019; Yang et al., 2021). These systems can be applied to goal-oriented domains by adding "API call" actions, enumerating database output as a sequence of tokens. Eric et al. (2017) use copy-augmented networks and the RNN learns to manipulate entity values, by saving them in a memory and being able to retrieve them when needed. Output is produced by generating a sequence of tokens, which can also come from a memory or by ranking all possible surface forms and picking the one with the highest probability. Hybrid methods, such as the Hybrid Code Networks (HCNs) (Williams et al.,

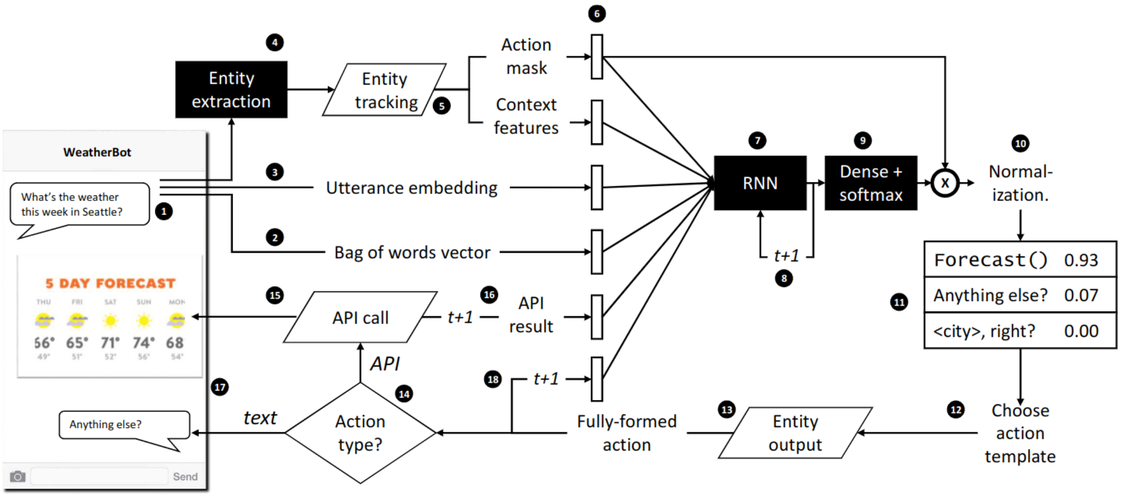


Figure 2.2: An overview of the Hybrid Code Networks architecture. Trapezoids refer to programmatic code, and shaded boxes are trainable components (Williams et al., 2017).

2017), try to combine the best of both worlds (Figure 2.2 shows the architecture).

HCNs use an RNN to accumulate dialogue state and choose actions. However, HCNs differ in that they use hand-crafted action templates, which can contain entity references, i.e. “Do you mean $< city >$?”. This design reduces learning complexity and also enables the software to limit which actions are available via an action mask, at the expense of human effort. In addition, HCNs are trainable via Reinforcement Learning (RL) as well, in comparison to past end-to-end recurrent models, which can be trained using supervised learning alone.

2.1.2 Incremental Dialogue

Data-driven goal-oriented dialogue systems are generally turn-based and do not support natural, everyday incremental dialogue processing. Such systems often cannot process naturally occurring incremental dialogue phenomena, such as mid-sentence restarts and pauses, or sub-dialogues (Howes et al., 2009). Incremental dialogue systems, instead of being turn-based, process the dialogue word-by-word, have shown reduced latency and to be beneficial and more natural for users (Skantze and Hjalmarsson, 2010). Incremental systems express the phenomenon of Compound Contributions (CCs), that is contributions in dialogue (and act as NLU parsers and NLG generators), which continue or complete earlier contributions, resulting in a single syntactic or semantic unit built across multiple contributions by one or more speakers (Purver et al., 2014). The incremental semantic parser for dialogue processing Dynamics of Language (DyLan) (Eshghi et al., 2011; Purver et al., 2011) implementation is an example of incremental formalism. Since incremental dialogue can be indefinitely extended, it naturally allows for representing incrementally growing meaning representations as more words are parsed or generated. Therefore, mod-

$$\begin{bmatrix} event & : & e_s \\ p1_{=today(event)} & : & t \end{bmatrix} \mapsto \begin{bmatrix} event_{=arrive} & : & e_s \\ p1_{=today(event)} & : & t \\ p2_{=pres(event)} & : & t \\ x_{=robin} & : & e \\ p3_{=subj(event,x)} & : & t \end{bmatrix} \mapsto \begin{bmatrix} event_{=arrive} & : & e_s \\ p1_{=today(event)} & : & t \\ p2_{=pres(event)} & : & t \\ x_{=robin} & : & e \\ p3_{=subj(event,x)} & : & t \\ x1 & : & e \\ p3_{=from(event,x1)} & : & t \end{bmatrix} \mapsto \begin{bmatrix} event_{=arrive} & : & e_s \\ p1_{=today(event)} & : & t \\ p2_{=pres(event)} & : & t \\ x_{=robin} & : & e \\ p_{=subj(event,x)} & : & t \\ x1_{=Sweden} & : & e \\ p3_{=from(event,x1)} & : & t \end{bmatrix}$$

“A: Today” ↦ *“..Robin arrives”* ↦ *“B: from?”* ↦ *“A: Sweden”*

Figure 2.3: Incremental interpretation via Type Theory with Records (TTR) subtypes (Eshghi et al., 2012), an extension of standard type theory used in semantics and dialogue modelling (Cooper, 2005).

elling dialogue based on explicit linguistic representations, such as a formal semantic grammar, can have an advantage for low-resource dialogue systems due to the incorporated prior knowledge (Shalyminov, 2020).

2.1.3 Pipeline, Incremental and End-to-end Methods

The aim of RQ1, is to address the low-resource expert emergency response domain with UXV and create a planning system that can converse with human experts. One challenge, if the aforementioned incremental models were to be used, would have been how to leverage the extra modalities, i.e. images and metadata of maps with GPS locations. Yu et al. (2016) explore integrating an incremental framework, i.e. DS-TTR (see Figure 2.3), with a set of visual classifiers, so as to ground the meaning representations produced during an interactive multi-modal dialogue system.

Advantages and disadvantages of pipeline and end-to-end methods (Chen et al., 2017; Shalyminov, 2020) are discussed below:

- Pipeline approaches have proven to be reliable, but they are difficult to scale to new domains. It is hard to manually encode all of the features and slots that might be used during a dialogue. Data-driven, end-to-end approaches, on the other hand, transcend these restrictions because all of their components are directly trained on conversational data, with no assumptions about the domain or dialogue state structure, allowing scaling up to new domains if the data is available. The ease of adaptation to new domains with different or similar input features is examined when answering RQ3.
- End-to-end based dialogue systems commonly rely on huge amounts of conversational annotated data and structured knowledge bases. These systems learn to talk by repeatedly replicating a response. Regardless of how effective that type of learning is, the responses may be bland and meaningless. A deeper awareness of the real world, as well as the capacity to ground concepts on that environment, is one method to help end-to-end agents learn more successfully.
- Trustworthiness can be an issue with end-to-end methods since their outputs can be quite unpredictable and their methods black boxes, compared to templated pipeline methods. Especially for domains, such as RAS, unexpected and unexplainable outcomes could be very costly. This is the reason why

in the industry they still rely mostly on pipeline or hybrid methods, instead of end-to-end for goal-oriented dialogue (Williams et al., 2017; Hastie et al., 2018a).

2.2 Situated Dialogue

In situated dialogue, the environment of interaction may be perceived differently from human to human, meaning that these referring expressions need to be carefully selected and verified, especially if the shared environment is ambiguous (Fang et al., 2013). Situated dialogue covers multiple aspects of interaction. These include: situated natural language processing (Bastianelli et al., 2016); situated reference resolution (Misu, 2018); language grounding (Johnson et al., 2017); visual question answer/visual dialogue (Antol et al., 2015); dialogue agents for learning visually grounded word meanings and learning from demonstration (Yu et al., 2017); and natural language generation (NLG), e.g. situated instructions and referring expressions (Kelleher and Kruijff, 2006; Byron et al., 2009).

RQ2 focuses on situated instructions, utterances in high-level natural language that are interpreted within a real-world context. Words in these utterances are symbols representing real objects, spatial relationships and actions on a specific context. The ability to associate language with real-world context (such as, objects, events, and actions) allows language users to work collaboratively on tasks, establish shared beliefs about the environment, and learn from others' experiences through linguistic communication.

However, due to the symbol grounding problem (Harnad, 1990), it has been a challenge that is yet to be solved without making simplifying assumptions. How do symbols (words, pictures) get their meanings? How can the meaning interpretation of a symbol system be made intrinsic to the system, rather than just a set of meanings for parroting in our heads?

The grounding problem can be categorised into two challenges: (1) the agent needs to learn to ground (map) Natural Language (NL) symbols onto their existing perceptual and lexical knowledge, such as a dictionary of pre-trained classifiers (Silberer and Lapata, 2014; Thomason et al., 2016; Matuszek et al., 2014); (2) the agent must learn both the perceptual categories themselves and also how NL expressions map to these without any prior knowledge of perceptual categories (Skočaj et al., 2016; Yu et al., 2016, 2017).

Bisk et al. (2020) define five levels of "World Scope (WS)", so as to examine whether meaning arises from the use of language to communicate, rather than the statistical distribution of words. **Corpus based representations (WS1)** is the scope in which words are represented by other words from a collected and annotated corpus. **The written world (WS2)**, extends this to every text that is accessible online. Despite the quantity of the data and advances in NLP algorithms (Devlin et al., 2019), Van Schijndel et al. (2019) analyse that text pretraining seems to have reached a plateau. Following, **vision and language (WS3)** includes learning from other forms of data, such as visual data. O'Grady (2005) and Vigliocco et al. (2014) pose that children require perception in order to learn language. Despite the fact that an agent in this scope can demonstrate better generalisation to WS2 agents, it will not be able to answer, for instance, the question "Which object will fall faster, 1g of a metal square or 1g of paper square?". To solve that, research on the next scope started growing, **embodied agents (WS4)**, that can act either in a virtual world (Tan et al., 2019; Bisk et al., 2018a) or the real world (Tellex et al., 2020) and can learn to generalise by acting in the world and getting "unlimited" feedback from the environment. Despite the complexity of the aforementioned WS, their training data does not provide signals hypothesising people's beliefs, perceptual abilities, and mental states in order to reduce perplexity or increase accuracy (DeVault et al., 2006). **The social world (WS5)** entails participating in linguistic activities, such as negotiation (Lewis et al., 2017) and collaboration (Chai et al., 2016), to learn about the influence language has on the world. Following the world scopes, Bisk et al. (2020) claim that language is not learnt just from text found on the internet (WS2), or just from videos (WS3) or just by acting on our own (WS4), but through interpersonal communication as well (WS5).

That being said, in the emergency response domain with UXV, described in this thesis, successful collaboration between human experts and the agent (WS5) is key. The environment of interaction is highly complex and dynamic, meaning that without the right inference, language grounding would be error-prone and could have serious consequences. Multi-modality is a necessity in this particular Command and Control (C2) task, since without visual stimuli of the scenario at hand, communication would be challenging. The goal of RQ1 and RQ2 is to address the complexity of tasking UXV, thus understanding the semantics of spatial relations and grounding expert language correctly via multi-modal signal (WS3). Surpassing the SOTA in language grounding is not the direct goal but rather a biproduct of making an NLP algorithm for fully autonomous collaboration between human experts and CA (WS5).

2.2.1 Grounding for Remote Vehicle Control

Language grounding refers to interpreting language in a situated context and includes collaborative language grounding towards situated human-robot dialogue (Chai et al., 2016), city exploration (Boye et al., 2014), as well as following high-level navigation instructions (Blukis et al., 2018). Lemon et al. (2001) present the WITAS dialogue system as introductory research for multi-modal conversations with autonomous mobile robots. They argue that human-robot interaction raises new challenges in comparison to previous work in dialogue systems in the travel-planning domain, in that the system must be mixed-initiative, asynchronous, open-ended, and involve a dynamic environment. Their dialogue manager is handling multi-modal input from the user, such as gestures, and UAV reports and questions, as well as sending speech and graphical outputs to the operator. However, the demonstrated system is strictly rule-based which limits generalisability and, most importantly, the understanding of maps and grounding natural language instructions to maps is out of the scope of that research. This thesis is concerned with situated instruction understanding for controlling remove vehicles, with RQ2 and RQ3 in particular, aiming at: 1) reading maps/charts and understanding natural language instructions, so as to plan missions for UXV; 2) being generalisable towards different tasks and environments (maps that do not belong in the nautical domain).

Mapping instructions to low-level actions has been explored in structured environments by grounding visual representations of the world and text onto actions using RL methods (Misra et al., 2017; Xiong et al., 2018; Huang et al., 2019). Works of this nature have been extended to controlling RAS through natural language instruction in a 3D simulated environment (see Figure 2.4) (Ma et al., 2019; Misra et al., 2018; Blukis et al., 2019), mixed reality (Huang et al., 2019) and using imitation learning (Blukis et al., 2018; Shridhar et al., 2021a). These methods attempt to predict a goal location and generate actions that control a UAV, given a natural language instruction, a world representation and/or robot observations. These prior works use raw pixels to generate a persistent semantic map from the system's line-of-sight image. In the domain of AUVs, usually the environment of interaction is a map/chart, such as Electronic Navigational Charts (ENC), and maps are compilation of structured metadata. A question arises, relevant to RQ2, on whether the structure of these data can be leveraged, combined or standalone with the raw pixels of the maps, when grounding natural language instructions to the world of interaction.

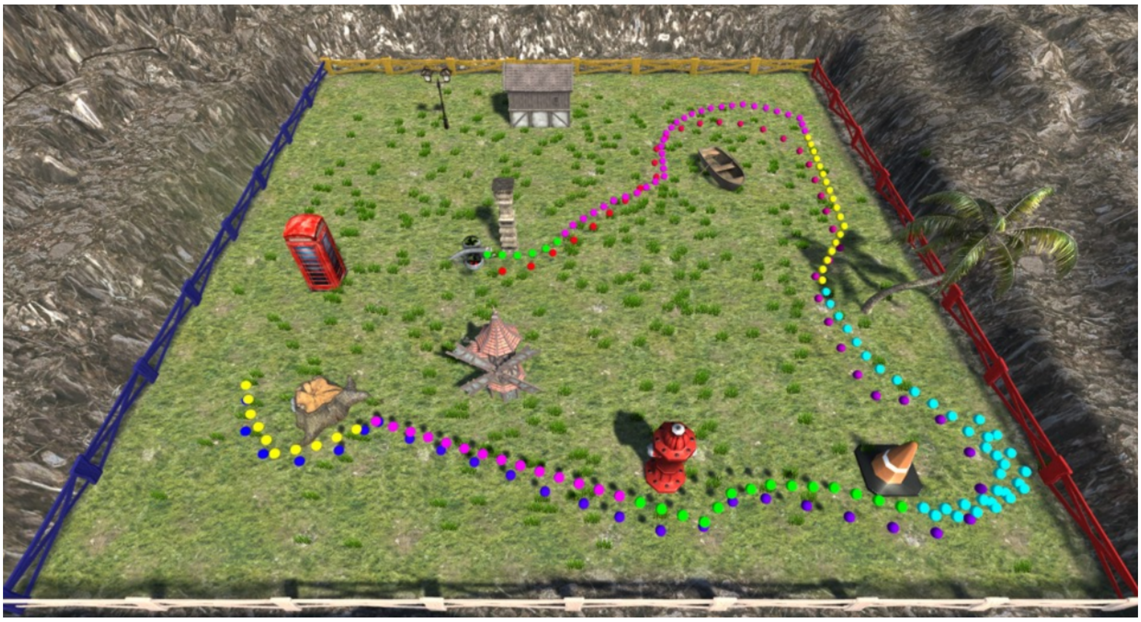


Figure 2.4: Example of mapping instructions to actions from the 3D simulated environment LANI (Misra et al., 2018). The agent Excerpt of the instructions segmented with colours: "Go around the pillar on the right hand side and head towards the boat, circling around it clockwise....".

Hemachandra et al. (2014) explore an algorithm that learns human-centred environment models by combining natural language descriptions with image/laser-based scene classification. The representation of topological and semantic maps is done through semantic graphs. Probabilistic graphical models have been used extensively for incorporating information about the semantics of the environment (Persson et al., 2007; Tellex et al., 2011a).

Tellex et al. (2011b) used a syntactic parsing approach that learns from a corpus of data, so as to construct a graphical model that grounds language into actions executed on a robotic forklift, e.g. move, pickup. Another interesting approach using graphical models is presented by Kollar et al. (2010), where they propose a system that predicts the most probable path through the environment by following natural language directions and extracting a sequence of "spatial description clauses" from the input and, using only information from the environmental geometry and detected visible objects

Blukis et al. (2018) tackle understanding navigational instructions using imitation learning. Key to their approach is building a semantic map of the environment within a neural network model. Their semantic map, a differentiable 3d vector, is part of a bigger neural network architecture and can save observed locations in the world in feature vectors. This approach comes with the advantage that it does not require maintaining a distribution over similar maps.

2.2.2 Grounding Referring Expressions

Referring Expressions (RE) can be noun phrases of any structure, such as 'cup on the table' in "Bring me the cup on the table" (Mao et al., 2016; Xu et al., 2015a; Cirik et al., 2018). The scope of this thesis is not to study RE but to study situated natural language instructions that include RE, to help locate objects from input images or maps. In grounding, when agents need to communicate for an emergency, RE can be crucial for coming up with the right strategy.

Grounding referring expressions is widely studied in Natural Language Processing, Computer Vision, and Robotics (Clark and Brennan, 1991; Pateras et al., 1995; Tellex et al., 2011b; Zhang et al., 2018a; Sadhu et al., 2019; Thomason et al., 2019b). Li et al. (2016b) identify four main issues in grounding for human-robot collaborative interaction:

Visual Search: Defined as the standard visual behaviour to find one object in a visual world filled with other distracting items (Wolfe, 1994). Firstly, the model processes information about basic visual features, i.e. colour, size, various depth cues, etc., in parallel across large portions of the image and secondly performs more complex operations, i.e. image recognition, reading, object detection etc., over smaller portions of the image.

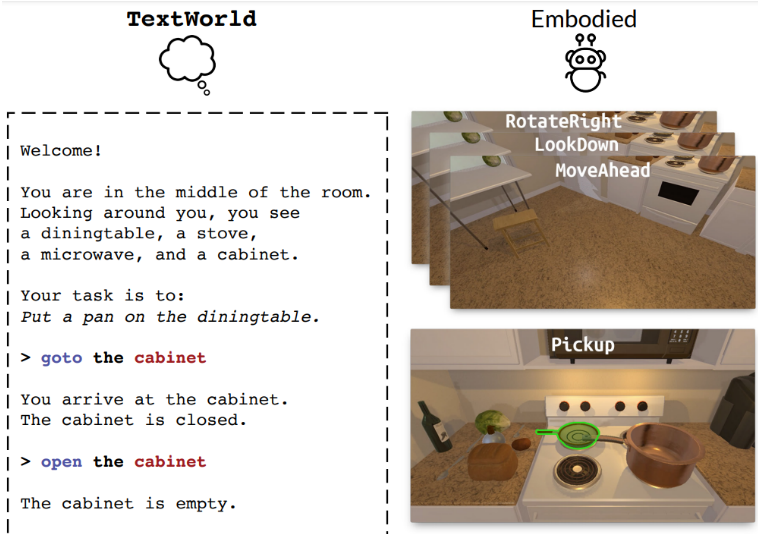
Spatial Referencing: Identifying the spatial orientation of an object when similar objects are involved. Moratz et al. (2001) refers to a group of similar objects as a whole to specify the target. Landmarks are unique objects at fixed locations and can be used as well when referencing a target in a navigation system, as well as, routes as a fixed sequence of locations to be used (Werner et al., 1997).

Perspectives: Taking each other's perspectives when referring to objects in a shared environment is crucial and challenging for the successful collaboration on spatial tasks (Franklin et al., 1992). Bloom et al. (1999) separate perspectives into three categories, i.e. deictic, intrinsic and absolute perspective, meaning, users' points of view, objects' points of view and referring to the world frame, respectively.

Ambiguity: When the participants are not explicit about the perspectives they are taking, instructions become unclear because assumptions would have to be made (Bloom et al., 1999). Further ambiguities that make the instruction harder to comprehend occur in the objects, landmarks and spatial relationships between them.

The work presented in this thesis, and specifically on RQ2 and RQ3, focuses on visual and language grounding, which can be considered a superset of grounding RE. It is thus important to be able to understand visual and language inputs, as well as generalise to unknown settings with different input elements or output goals. Visual grounding of RE is closely related to object recognition, as a classification task, with predetermined set of object categories. Previous work learns to comprehend natural language object descriptions for unrestricted object categories by using convolutional neural networks (CNN) instead of handcrafted visual feature extractors and recurrent neural networks (RNN) instead of language parsers (Mao et al., 2016). These networks are capable of connecting automatically visual and language concepts by jointly embedding them in an abstract space. However, most of these methods use the entire image and recently successful work is proposed that, instead of modelling the entire context, explicitly models the referent and context region pairs (Nagaraja et al., 2016; Zhang et al., 2018b).

Region Proposals: Instead of using the output vector of a CNN to represent the entire image, the task is converted to that of object detection and the features of detected objects are used as the embeddings of the image. An object detector, such as Faster R-CNN (Ren et al., 2015a) detects $\mathbf{k}$ objects $\{\mathbf{o_1}, ..., \mathbf{o_k}\}$ from the image, for example in Figure 2.6, where each object is represented by its 2048-dimensional Region-of-Interest (RoI) feature vector and its position feature bounding box.



```
Welcome!

You are in the middle of the room.
Looking around you, you see
a diningtable, a stove,
a microwave, and a cabinet.

Your task is to:
Put a pan on the diningtable.

> goto the cabinet

You arrive at the cabinet.
The cabinet is closed.

> open the cabinet

The cabinet is empty.
```

Figure 2.5: Combining text and physical actions, left and right respectively, in ALFWorld (Shridhar et al., 2021b).

Domains that combine language and the world (see Figure 2.5) have gathered attention recently (Shridhar et al., 2021b; Anderson et al., 2018). In terms of the method used to tackle the challenge of grounding language to vision tasks, attention mechanisms (Bahdanau et al., 2015; Vaswani et al., 2017a; Xu et al., 2015b) have proven to be very robust and here inspiration is drawn from the way Xu et al. (2015b) use attention to solve image captioning by associating words to spatial regions within a given image. Recent advances in visual grounding can be split into two categories:

One-stage methods (Xinpeng et al., 2018; Sadhu et al., 2019; Yang et al., 2019b; Liao et al., 2020) do not generate any region proposals for vision features as in the two-stage approach below, but fuse densely the linguistic with the visual features before leveraging the language-attended feature maps to perform bounding box prediction. Real time Cross-modality Correlation Filtering method (RCCF) (Liao et al.,

2020) reformulates the visual grounding problem as a correlation filtering process. This is done by firstly grounding the language to the visual domain and then, as a template, performing correlation filtering on the image feature map. It then picks the peak value of the correlation heatmap as the centre of target objects. It runs at 40 frames per second (FPS) and achieves leading performance in different benchmarks and almost doubles the SOTA performance on the RefClef (Kazemzadeh et al., 2014a).

Two-stage methods (Wang et al., 2017; Tan and Bansal, 2019; Lu et al., 2019; Chen et al., 2020; Zhuang et al., 2018; Zhang et al., 2018a; Yu et al., 2018) firstly generate the aforementioned region proposals of the image, as features instead of the pixels, and then leverage the language expression to select the best matching region in the second stage. The region proposals are generated using either unsupervised methods (Plummer et al., 2018; Wang et al., 2017) or a pre-trained object detector (Yu et al., 2018; Tan and Bansal, 2019). Studies, such as the UNITER (Chen et al., 2020), LXMERT (Tan and Bansal, 2019) and VisualBERT (Li et al., 2019), achieve SOTA results on challenging datasets for joint reasoning about natural language and images such as the NLVR2 (Suhr et al., 2019b). ViLBERT (Lu et al., 2019) and LXMERT (Tan and Bansal, 2019) introduced the two-stream architecture, where two Transformers are applied to text and images independently first, and then fused by a third Transformer (see Figure 2.6).

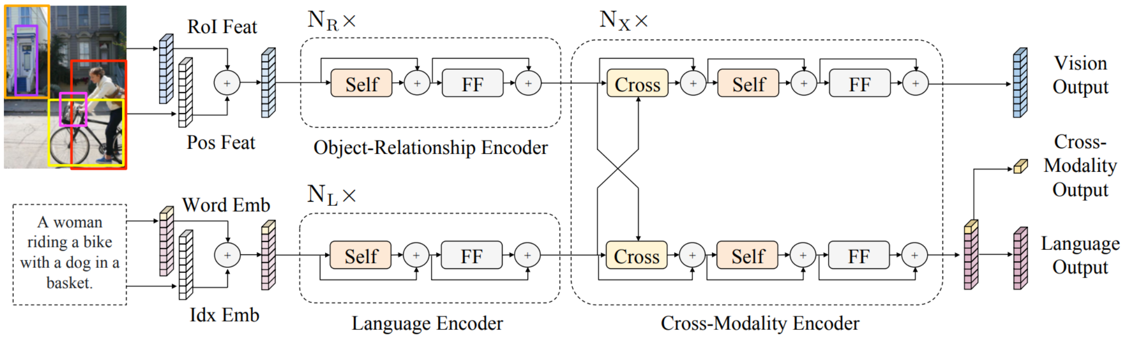


Figure 2.6: LXMERT's structure of transformer encoder layers for learning vision-and-language cross-modality representations (Tan and Bansal, 2019).

2.2.3 Transformers for Visual Grounding

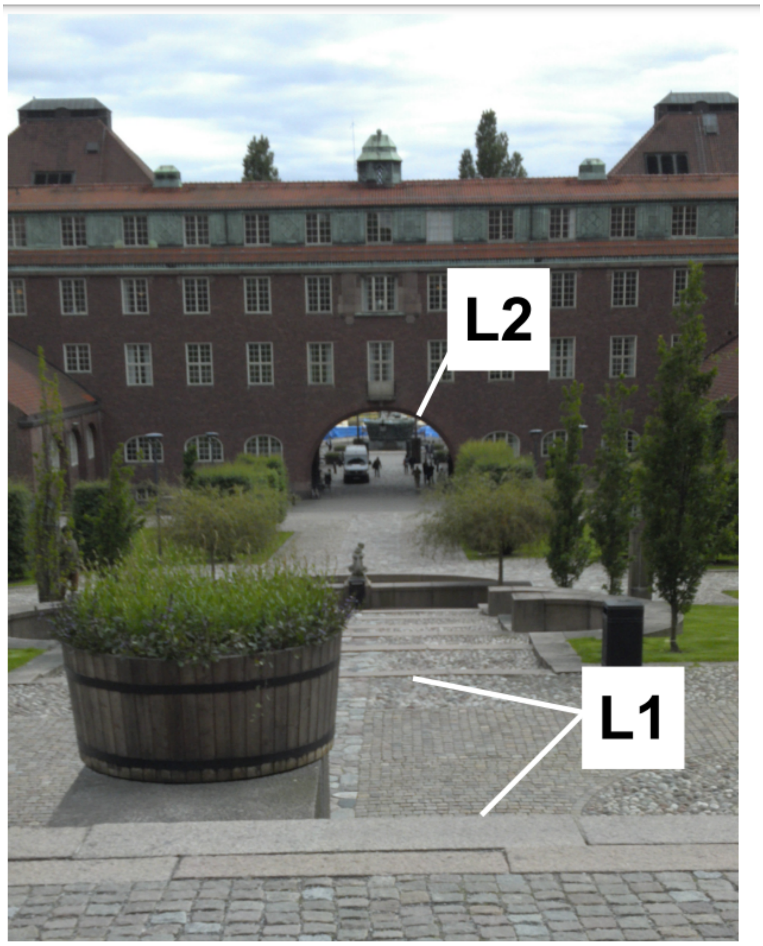
Most, if not all, SOTA methods for visual grounding use stacked transformers. The transformer is first introduced in (Vaswani et al., 2017b) to tackle the Neural Machine Translation (NMT). The primary component of a transformer is the attention module, which is a mapping of a query and a set of key-value pairs to an output as a weighted sum of the values. Given the trained weights of the key-value pairs, the product masks out the "irrelevant" information and learns to aggregate only the important ones. Compared to recurrent architectures (Hochreiter and Schmidhuber, 1997; Mikolov et al., 2010), the attention mechanism exhibits better performance in processing long sequences. Inspired by the success of transformers in NMT, a series of transformers (Carion et al., 2020; Chen et al., 2021; Kolesnikov et al., 2021; Yang et al., 2020; Zeng et al., 2020; Zhu et al., 2020) applied to vision tasks have been proposed. The DEtection TRansformer (DETR) (Carion et al., 2020) replaces the hand-crafted object detection pipeline with a transformer and approaches object detection as a direct set prediction problem. It showcases a fixed set of learnable object queries, reasons about global context and about the relations of the objects, so as to directly output the final set of predictions in parallel, which makes it faster and more efficient. Motivated by BERT (Devlin et al., 2019), which stands for Bidirectional Encoder Representations from Transformers, the research community started to investigate vision-language pre-training (VLP) (Chen et al., 2020; Lu et al., 2019; Tan and Bansal, 2019; Su et al., 2020; Li et al., 2020) to jointly represent text and images. Generally, these models take as input features, the region proposals and text, and use several transformer encoder layers for joint representation learning.

2.2.4 Instruction Understanding

Situated instruction understanding is a wide and necessary task for the successful collaboration between humans and agents. Instructions of this kind need to be accurately grounded in the shared environment. In the emergency response domain, since the environment is structured maps, the task can be translated to mapping natural language to map structures that are comprised of GPS coordinates. With respect to RQ2, the aim is to ground instructions in electronic maps, such as OSM, and predict the right GPS locations of grounded objects on the map. There is not much previous work that attempts this challenging task. Brébisson et al. (2015), as part of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD) challenge in 2015, use Neural Networks for Taxi Destination Prediction as a sequence of GPS locations. However, this does not involve natural language instruction processing.

SPACEREF (Götze and Boye, 2016) is a corpus of street-level geographic descriptions, collected using the Wizard of Oz (WOz) methodology (Dahlback and Jonsson, 1989). The participants' purpose was to walk and reference as many visible landmarks to the system as possible. The participants were given an unlabelled map with no common symbols or names, as well as a course to follow. Their resulting issues include learning from the use of landmarks to determine landmark salience and resolving references in a dynamic environment. For referencing expression resolution, OSM metadata was used to determine the exact pedestrian's GPS position and identify what landmarks were available in the given location. Furthermore, their linguistic instructions differ in that they do not require long distances or 2D maps with identified objects. SPACEREF is perhaps the nearest to the task of GPS prediction, in that the task entails both GPS tracks from OSM and annotated mentions of spatial entities in natural language (see Figure 2.7). Nevertheless, it is different from the task surrounding RQ2 in that the perspective of these spatial entities is the first-person and are viewed and referred to as such, rather than entities on a map, e.g. first-person view, example from SPACEREF: “I continue in a southwesterly direction down the steps ” vs third-person view: “Move a husky robot 300m southwest of the Bay Regional Shoreline” from the dataset (ROSMI) introduced for addressing RQ2.

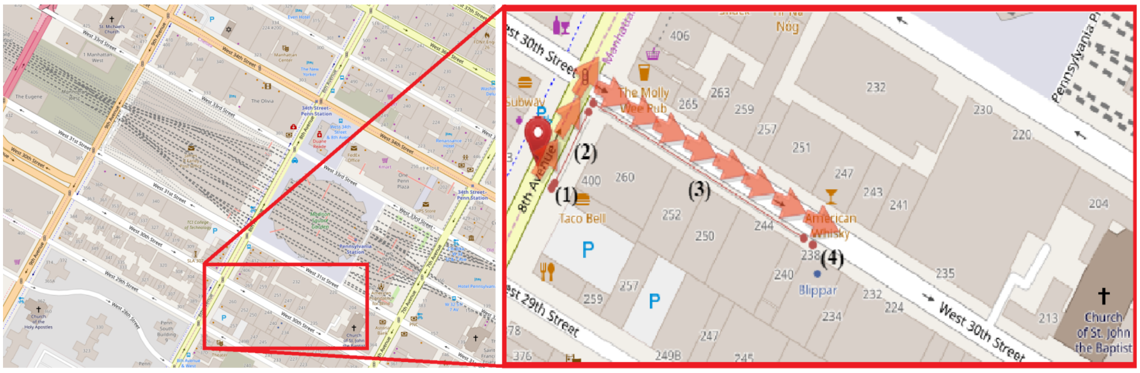
PURSUIT (Blaylock, 2011), a geospatial path descriptions corpus, was collected with the aim of tackling spatial orientation, locations, movement and paths by understanding natural language descriptions that are grounded in GPS locations. Audio recordings were collected while driving in an urban area in real time, accompanied by the GPS data of the crossed path. In spite of the inclusion of OSM on the aforementioned tasks, neither SPACEREF nor PURSUIT, incorporate any visual data that can be leveraged for training joint vision and language models. Moreover, the nature of their task is essentially different from the task of emergency



‘I continue in a southwesterly direction *down the steps* [L1] *towards the arch at the bottom* [L2]’

Figure 2.7: Example utterance and image from SPACEREF (Götze and Boye, 2016).

response with UXV that is explored throughout this thesis. Paz-Argaman and Tsarfaty (2019) introduce a new task called Realistic Urban Navigation (RUN), based on OSM. The goal is to contribute in understanding navigation instructions grounded on a real urban map (see Figure 2.8). The instructions are given in the form of a paragraph. They illustrate complexity in the possible compounded errors that may arise from the long routes, since the complete route is evaluated and not just the final position. They also allow OSM's rich environment with varied entities (including most unseen during training). Paz-Argaman and Tsarfaty (2019) propose an encoder-decoder model to output the right sequence of actions, i.e. turn, walk, end. Their novel components include the “Entity Abstraction” which learns to replace out-of-vocabulary words from the instruction, with variables (e.g. “As you walk towards Startbucks, turn right” becomes “As you walk towards Y_1, turn right”) and then a “World-State Processor” that takes the entities from the map's metadata and tries using attention to map them with the right variables before passing the hidden state to the decoder, which is an LSTM. Below we expand the perspective on works



Instructions: (1) As you walk out of Taco Bell on 8th Avenue, turn right. (2) Then turn right as you reach the intersection of West 30th Street. (3) Now head down West 30th Street for approximately a half block. (4) You have gone too far if you reach Church of St. John the Baptist.

Figure 2.8: Example of the RUN task. Above, the whole map and a part of it, left and right respectively, and navigation instructions below. (Paz-Argaman and Tsarfaty, 2019).

that do not necessarily entail GPS locations, but describe different environments and datasets on the task of vision and language navigation (VLN). The works are split in single- and multi-turn instruction navigation.

2.2.5 Single-turn Instruction Navigation

Misra et al. (2018) collected a corpus of navigation instructions execution using the LANI simulator and crowdsourcing. They create environments at random and create a single reference path for each one. The execution of instructions was broken down into two parts: goal prediction and action generation. To make a map from visual input to objectives output, they introduced *LINGUNET*, a new language-conditioned image generating network architecture. In addition, they test their method on the CHAI corpus, a household instruction execution task on the CHALET (see Figure 2.9) simulated environment (Yan et al., 2018).

Before LANI, Blukis et al. (2018) proposed a virtual environment navigation task based on the UnrealEngine, which uses AirSim (Shah et al., 2018) to simulate realistic quadcopter dynamics. The agent is a quadcopter that hovers around landmarks. The problem of mapping, planning, and task execution of language instructions created from a pre-defined set of templates was the focus of their research. Each command is placed into an LSTM to obtain the language embedding, and each

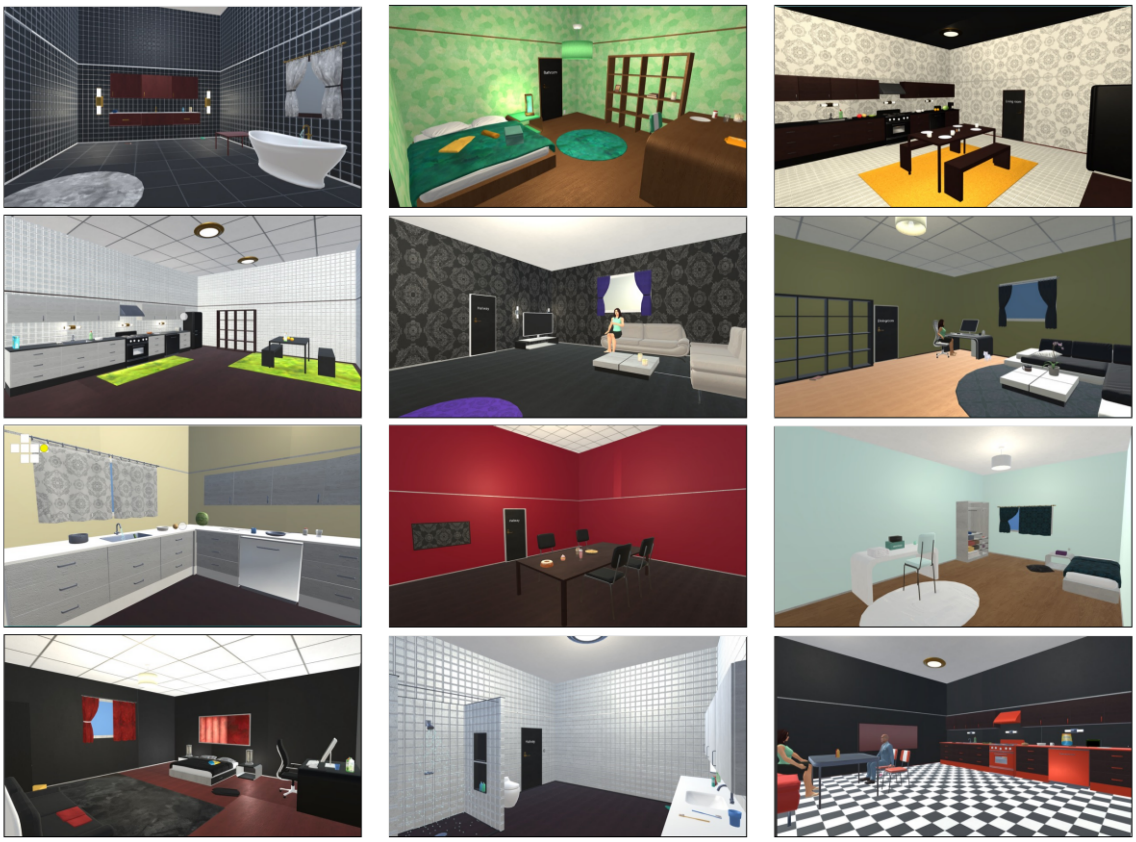


Figure 2.9: Examples of numerous rooms from CHALET. (Yan et al., 2018).

image is placed into a customised residual network to represent it at each time step in Blukis et al. (2018) baseline technique.

Blukis et al. (2019) suggested a combined, supervised and RL, learning framework to map language and images to low-level action output called, Supervised Reinforcement Asynchronous Learning (SuReAL), so as to combine the simulation and reality. It can be used, both in simulation and reality, without the need to fly in the real world during training. Last, but not least, in a more recent study, Blukis et al. (2020) extend the problem to reason about new objects. Nevertheless, due to the lack of sufficient training data, in order to ground objects with the correct mentions in the language instruction, they used a few-shot method trained from extra augmented reality data.

LANI and CHAI, both from virtual environments, are grounded in a scenario observed by the robot rather than a 2D map. This type of real-time streaming of the first-person perspective may not be available in the emergency response area described in this thesis, hence we intend to rely solely on map-based interaction (RQ2).

Wu et al. (2018) proposed an environment House3D, and based on that, devel-

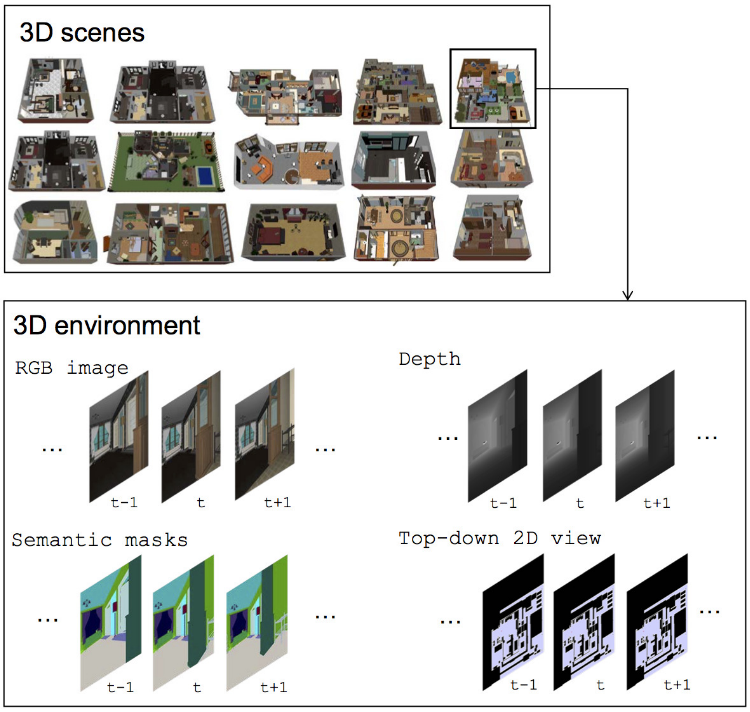


Figure 2.10: Examples of the House3D environment, which consists of thousands of indoor scenes from the SUNCG dataset (Zhang et al., 2018c).

oped the Concept-Driven Navigation, called RoomNav (see Figure 2.10) and sourced from the SUNCG dataset (Zhang et al., 2018c). The instructions are of the form "Go to X", where X represents a pre-defined room type or object type. Besides the goal instruction, each scene is provided with three different visual input data: (1) raw pixel; (2) semantic segmentation mask of the pixel input; and (3) depth information. This is a semantic concept, and the agent needs to ground the instruction to a variety of scenes with different visual appearances. For tackling this challenge a gated-CNN and gated-LSTM network are used for controlling continuous and discrete actions respectively.

There is also a lot of work being done on mapping natural language commands to actions in a 3D world for robots, such as Room-to-Room (R2R) navigation (see Figure 2.11) for visually-grounded natural language navigation (Anderson et al., 2018).

Based on the R2R task (Anderson et al., 2018), Chi et al. (2020) developed an interactive learning framework to allow the agent to ask for help when needed and are among the first to introduce human-agent interaction in the instruction-based navigation task. Their base model architecture is inspired by previous work in VLN (Tan

Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at pictures and table. Wait by the moose antlers hanging on the wall.

Figure 2.11: Example of the Room-to-Room (R2R) navigation task. (Anderson et al., 2018).

et al., 2019), with the instruction being encoded using a bidirectional LSTM and the attentive panoramic view, which serves as the visual input with a ResNet (He et al., 2016). Two ways are presented for executing actions given the inputs: (1) a simple confusion-based (MC) model, in which the agent should ask a question if it is unsure of itself, i.e. if the predicted action distribution is not sharp; (2) a more sophisticated method that demonstrates reasoning, in which the agent discovers the best timing and locations to ask questions using RL with reward shaping (ASA). They demonstrate that the RL agent can dynamically react to noisy human replies. Finally, they offer a data-efficient and realistic continuous learning technique for fine-tuning the agent using augmented data obtained from the human-agent interaction. Similarly, data augmentation from data collected from human-agent interaction is addressed in RQ4 together with the exploration of HITL methods for allowing an NLP algorithm to continually improve without data expert intervention. When the trained agent is being tested, the answers that are provided assist in correcting the wrong trajectories and the corrections then serve as augmented data to prevent the agent from repeating the same mistakes. However, their method relies on imitation learning techniques and purely on data augmentation, rather than the complete HITL framework that involves learning through direct collaboration.

Task-driven Embodied Agents that Chat (TEACh) (Padmakumar et al., 2021) is one of the first datasets where human-human (see Figure 2.12), conversational dialogues were used to perform object interaction, such as picking up an object, and state changes, such as slicing bread, in a visual simulation environment. R2R, on the other hand, is not conversational and does not allow object interaction or state changes. TEACh involves a human *Follower* who engages in free-form, rather than turn-taking, dialogue with a human *Commander*. Both Follower and Commander collaborate in natural language to complete household tasks in the AI2-THOR simulator (Kolve et al., 2017). They establish baseline performance using a modified Episodic Transformer (ET) (Pashevich et al., 2021), designed for the ALFRED benchmark (Shridhar et al., 2020). ET comprises of a transformer language encoder for encoding the instructions, uses a ResNet (He et al., 2016) for encoding visual

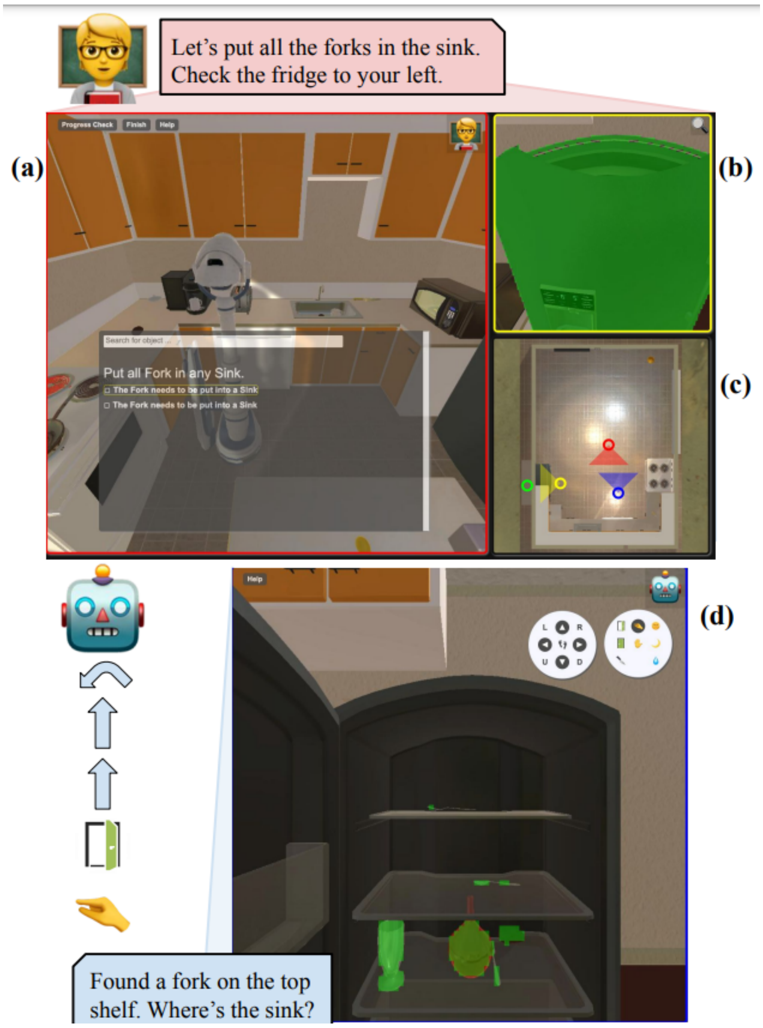
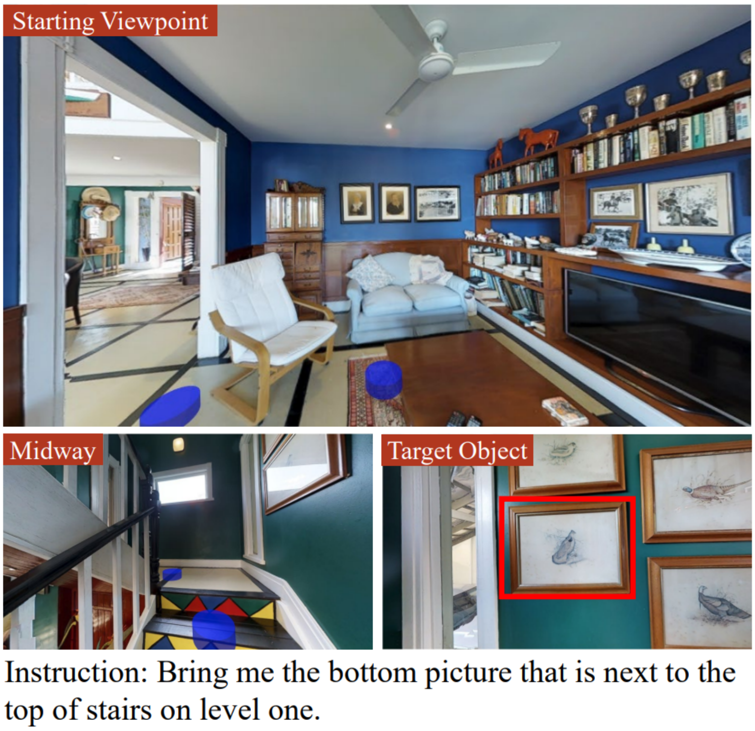

Let's put all the forks in the sink.
Check the fridge to your left.
(a)
Progress Check
Finish
Help
Search for object ...
Put all Fork in any Sink.
□ The Fork needs to be put into a Sink
□ The Fork needs to be put into a Sink
(b)
(c)
Help
(d)
Found a fork on the top shelf. Where's the sink?

Figure 2.12: Example of TEACh dataset, with the Commander and Follower interacting (Padmakumar et al., 2021), based on the AI2-THOR simulator (Kolve et al., 2017).

observations and two multi-modal transformer layers, so as to fuse information from the language, image, and action embeddings, followed by a fully connected layer to predict the next action and target object category. They modify ET by learning a new action prediction head for TEACh actions.



Instruction: Bring me the bottom picture that is next to the top of stairs on level one.

Figure 2.13: Example of the photo-realistic 3D environment REVERIE (Qi et al., 2020).

Remote Embodied Visual referring Expression in Real Indoor Environments (REVERIE) (Qi et al., 2020) task requires the agent to correctly locate a remote target object (see Figure 2.13), which is not visible at the starting position, specified by the high-level natural language instructions. Therefore, the agent needs to first navigate to the target location. Success in this task requires navigating through an unseen environment to identify an object, posing a practical challenge that reflects core visual problems in robotics. A task is considered successful if the right bounding box is selected from a set of candidates. Lin et al. (2021) introduce two pre-training tasks: (1) Scene Grounding task; (2) Object Grounding task. These pre-training tasks aid the agent to learn where to stop and what to attend to. In addition, they propose a new memory-augmented attentive action decoder that uses past observations to merge visual and textual information in an effective way. Despite their SOTA performance, the challenge of working in uncharted territory and how to reduce navigation time remains unsolved.

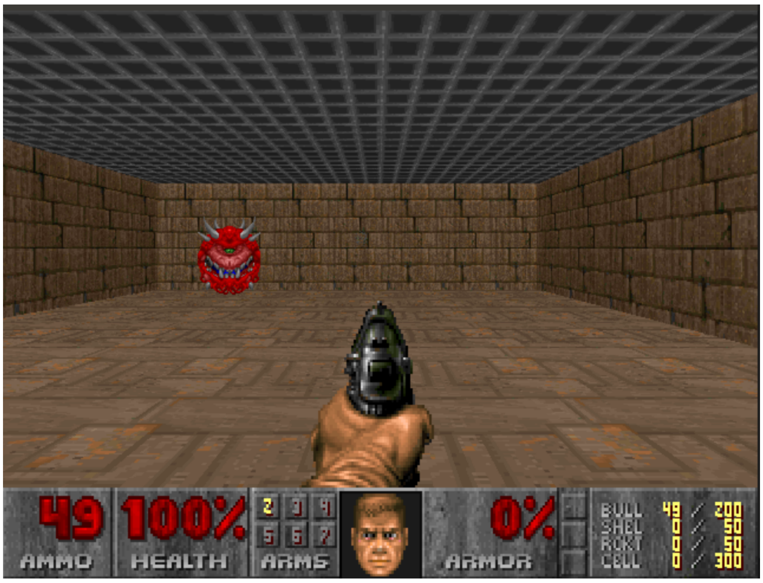


Figure 2.14: Image of the basic scenario scenario in VizDoom (Kempka et al., 2016)

Chaplot et al. (2018) constructed a language grounding environment in which the agent follows natural language instructions and is rewarded for performing the task correctly. The environment is built on top of VizDoom (Kempka et al., 2016), based on Doom, a first-person shooting game (see Figure 2.14). In the 3D Doom task, an instruction is a triplet of action, attributes, object. Each instruction can have up to one action and object, but they can have multiple attributes. This environment enables the creation of various objects with varying visual features at various points on the map. Success rate is defined by the successful navigation over the total number of tries. Chaplot et al. (2018) suggested a model that combines visual and text representations using a Gated-Attention mechanism, as well as RL and imitation learning approaches to learn strategies for implementing natural language instructions.

The navigation task introduced by Fu et al. (2019) is based on SUNCG (Song et al., 2017), in which an agent is given a location that corresponds to a room or object and must move through the house to find the target location. The language commands, of the form "go to X", were generated based on a preset grammar, with X representing the names of locations and things in the environment. Furthermore, Fu et al. (2019) proposed Language-Conditioned Reward Learning(LC-RL), which learns language-conditioned reward functions using Inverse Reinforcement

Learning (IRL), so as to ground free-form natural language commands in a high-dimensional visual environment.

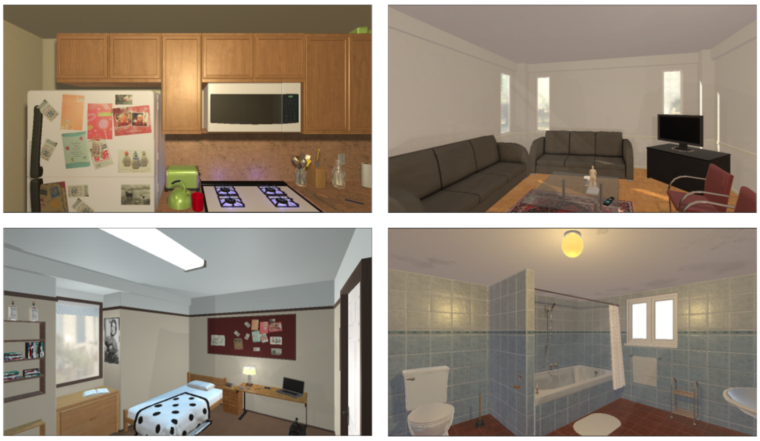
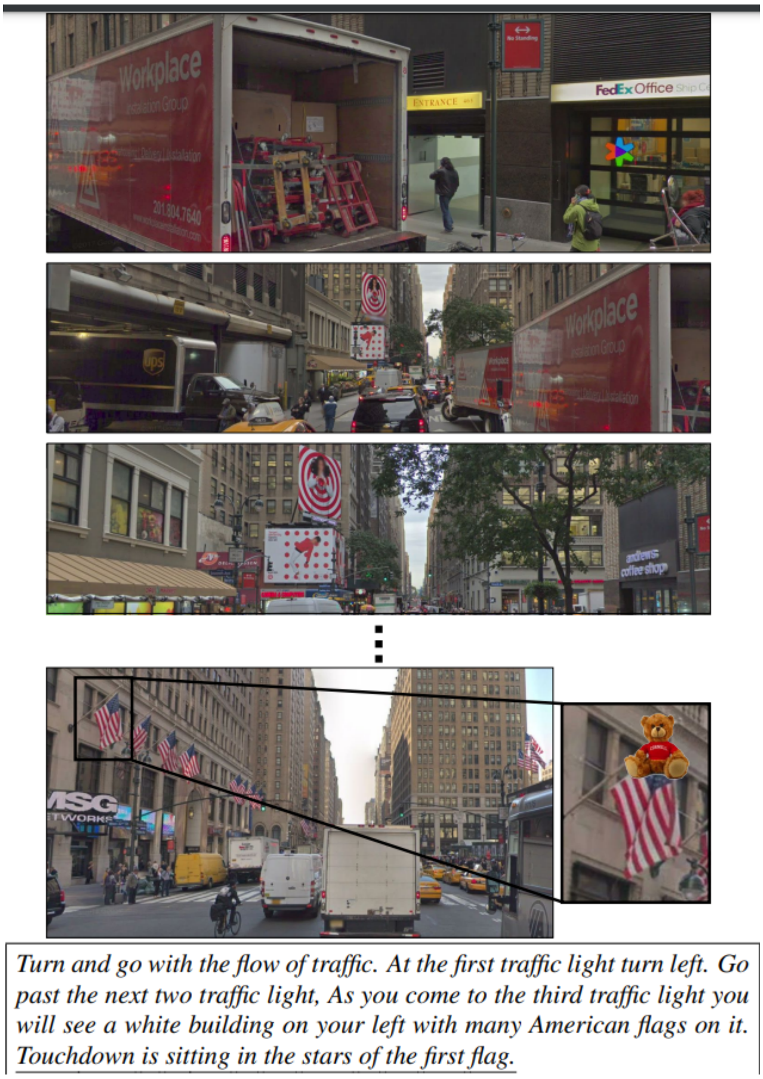


Figure 2.15: Example of AI2-THOR scenes (Kolve et al., 2017).

AI2-THOR (Kolve et al., 2017) is a large-scale photo-realistic 3D indoor corpus where agents can traverse and interact with items to complete tasks (see Figure 2.15). Deep reinforcement learning, planning, visual question answering, referencing expression resolution, object identification, and segmentation are all possible with AI2-THOR. Action Learning From Realistic Environments and Directives (ALFRED) (Shridhar et al., 2020) is a benchmark for learning a mapping from natural language instructions and egocentric vision to sequences of actions for home duties, built on top of AI2-THOR. When compared to other vision-language task datasets, ALFRED tasks have more complexity in terms of sequence length, action space, and language. In ALFRED, agents are given a natural language instruction about how to do a household chore, as well as, a first-person visual observation, and they must generate a sequence of actions. ALFRED has a total of 428,322 image-action pairs, with 25,743 English instructions explaining 8,055 expert examples, each with an average of 50 steps. These demos include a wide range of actions and partial observations. Min et al. (2022) propose a modular method with structured representations that: (1) processes natural language instructions into structured forms; (2) translates egocentric visual features into a semantic metric map; (3) predicts a search goal location using a Semantic Search Policy; and (4) outputs following navigation actions using a Deterministic Policy. Without the use of expert trajectories or low-level instructions, FILM achieves SOTA performance on ALFRED while consuming less data.

Visual Semantic Navigation task (Yang et al., 2019a) is based on the interactive environments of AI2-THOR (Kolve et al., 2017). From the 87 object categories within AI2-THOR, however only 53 categories are visible without any interaction. To evaluate the generalisation capacity of their strategy, these categories are further divided into known and novel sets. In training, just the known set of object categories is employed. For their experiments they used only the navigation commands of AI2-THOR (i.e. move forward, rotate right, move back, rotate left, and stop). This challenge is addressed by incorporating prior knowledge into a Deep Reinforcement Learning framework. Since graph structures represent how the information propagates between different nodes, Yang et al. (2019a) proposed to use Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017).



Turn and go with the flow of traffic. At the first traffic light turn left. Go past the next two traffic light, As you come to the third traffic light you will see a white building on your left with many American flags on it. Touchdown is sitting in the stars of the first flag.

Figure 2.16: Example interaction from the TOUCHDOWN (Chen et al., 2019) dataset.

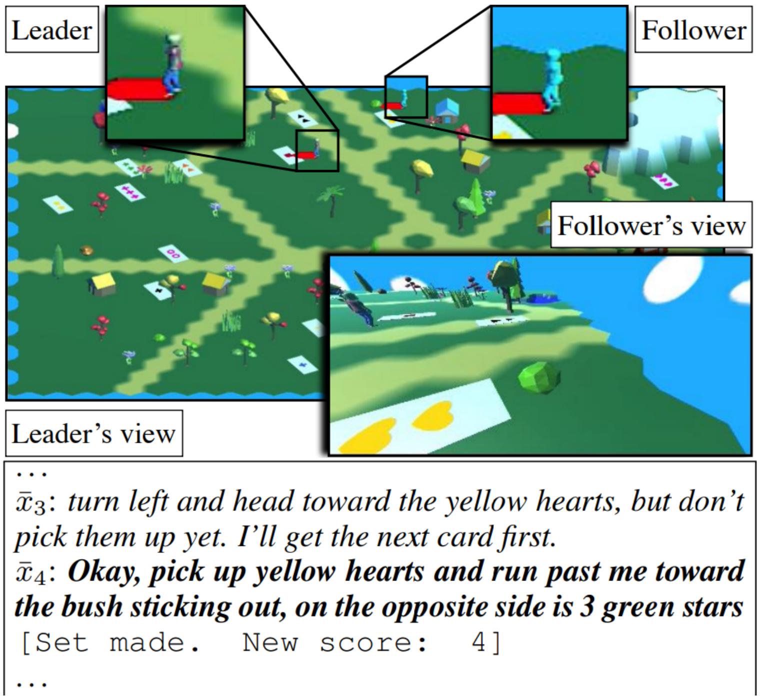
TOUCHDOWN, a dataset for natural language navigation and spatial reasoning based on real-life visual observations, was introduced by Chen et al. (2019) (see Figure 2.16). Existing works have focused on unrealistic and simple visual input, meaning that the diversity of the visual stimuli these environments provide is limited. Thereby, in TOUCHDOWN and other works, researchers focus on outdoor navigation based on Google Street View[1]. Two tasks are defined, that of navigation and spatial description resolution. When the agent walks to each accessible point connected to the undirected navigation graph, it obtains a 360° RGB panorama. New York City is represented by 29,641 panoramas and 61,319 edges in the environment. Xiang et al. (2020) achieve the SOTA on TOUCHDOWN by proposing a model that is Learning to Stop (L2STOP), a policy module that differentiates "STOP" and other actions, compared to most VLN models that treat all actions equally, which can have a great impact for this task since the STOP action terminates the episode and there is no chance for recovery, compared to the rest of the actions.

2.2.6 Multi-turn Instruction Navigation

There are tasks when the agent can only accept instructions, despite the collaborative aspect of the task itself. One example is the CEREALBAR (Suhr et al., 2019a), which simulates a scenario in which a leader and a follower work together to select cards in order to win points (see Figure 2.17). The players receive a point if they collect a valid collection of cards. The leaders' fully visible map environment allows them to design the path for the follower and give language instructions to guide them. The follower views the world from a first-person perspective and requires instructions to deal with the partial observability. The follower is unable to reply to the leader. Correct execution of instructions and the overall reward at the end are used to evaluate performance. Suhr et al. (2019a) offered a learning strategy focusing on recovering from compounding errors between consecutive instructions, as well as LINGUNET (Misra et al., 2018) as a modelling tool for explicitly reasoning about instructions with various aims, to address this challenge.

Humans, on the other hand, tend to make collaborative work more interactive

[1] https://developers.google.com/maps/documentation/streetview/overview



...

$\bar{x}_3$: *turn left and head toward the yellow hearts, but don't pick them up yet. I'll get the next card first.*

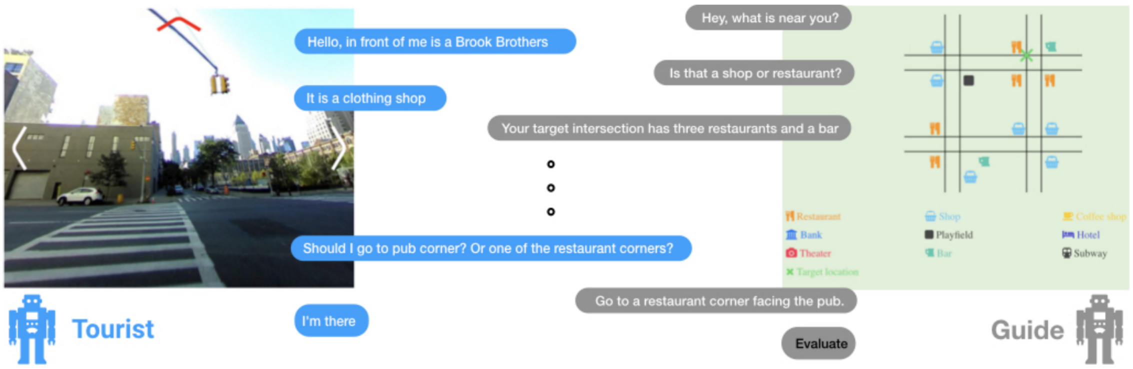
$\bar{x}_4$: ***Okay, pick up yellow hearts and run past me toward the bush sticking out, on the opposite side is 3 green stars***

```
[Set made. New score: 4]
```

...

Figure 2.17: Example interaction from the CEREALBAR (Suhr et al., 2019a) dataset.

than imperative, allowing both the "guide" and the "follower" to ask questions and contribute information. Vries et al. (2018) introduce a navigation task for tourist service and build a large-scale dialogue dataset, "Talk The Walk" (see Figure 2.18). The visitor is in a New York City 2D grid setting with 360-degree views of the



Hey, what is near you?

Hello, in front of me is a Brook Brothers

Is that a shop or restaurant?

It is a clothing shop

Your target intersection has three restaurants and a bar

Should I go to pub corner? Or one of the restaurant corners?

Go to a restaurant corner facing the pub.

I'm there

Figure 2.18: Example interaction from the Talk The Walk (Vries et al., 2018) dataset.

neighbourhood blocks. The guide has an abstracted semantic map of the blocks, and the destination position is shown to the guide unambiguously from the start. A guide's goal is to converse with a "tourist" in natural language in order to assist the latter in navigating to the correct spot. Vries et al. (2018) focus on tourist localisation and introduce a Masked Attention for Spatial Convolutions (MASC) mechanism to anchor the follower's utterances into the guide's map.

The Map Task Corpus (Anderson et al., 1991), looks at the value of referring objects when giving directions on a map, whether for identification or to show how well people understand their shared environment. In contrast to passively following instructions, the task of emergency response with UXV, regarding the RQ1, includes people collaboratively achieving a goal in a dynamic shared environment. Rather than simply discussing based on static maps, the environment is dynamically altered, with new items that are also movable, and the user and the agent collaborate to come up with the appropriate RE (e.g., "Let us use asset1 for target1"). Similarly, it has been done interactively by Schlangen (2016), where they ground non-linguistic visual information through conversation.

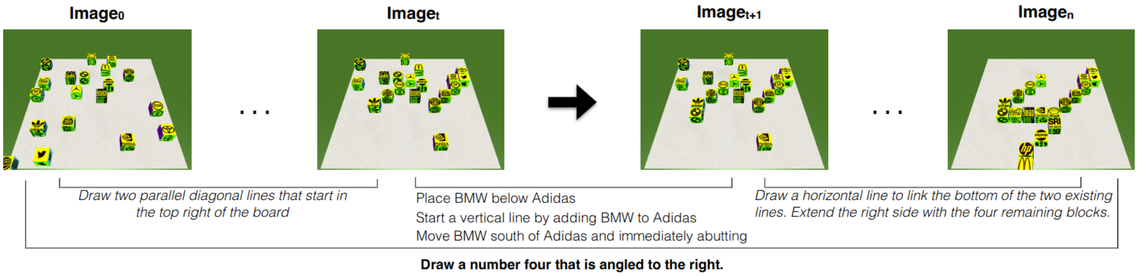


Image_0
Image_t
Image_{t+1}
Image_n
. . .
. . .
Draw two parallel diagonal lines that start in the top right of the board
Place BMW below Adidas
Start a vertical line by adding BMW to Adidas
Move BMW south of Adidas and immediately abutting
Draw a horizontal line to link the bottom of the two existing lines. Extend the right side with the four remaining blocks.
Draw a number four that is angled to the right.

Figure 2.19: Figure taken from the original paper (Bisk et al., 2016a) with descriptions of differing levels of abstraction (light,italic,bold) with a final goal of using the blocks to build a structure that looks like the digit four.

The Blocks World (Bisk et al., 2016a) task investigates understanding natural language instructions in a situated environment world and has been popular with many works (Misra et al., 2017; Bisk et al., 2018b; Tan and Bansal, 2018; Mehta and Goldwasser, 2019; Platonov et al., 2020). The task investigates understanding natural language instructions in a situated environment between a human and an agent. The human's goal is to arrange blocks in a certain way by giving instructions to the agent (see Figure 2.19). There are different versions of the dataset where blocks have digits, logos or are black blocks. The initial task was to draw digits taken from the MNIST (Lecun et al., 1998) dataset. Later Bisk et al. (2018b) extends their work to include rotations, 3D construction, and human created designs.

2.3 Human-in-the-loop Learning for Low-resource Domains

Deep learning (LeCun et al., 2015) methods are becoming important due to their ability of dimensionality reduction (Petscharnig et al., 2017) and demonstrated success at tackling complex learning problems (Zhang et al., 2015). Natural language is not always accurate; it is frequently ambiguous, and its structure is influenced by a variety of factors, including slang, regional dialects, and social context. Incorporating deep learning methods in NLP tasks and especially for feature representation can surpass many standard machine learning methods. The great success of deep learning is due to the larger models (Brutzkus and Globerson, 2019). However, due to the enormous number of parameters, a large amount of training data with labels is required (Zhou et al., 2014). In terms of size and density, the existing datasets are rapidly becoming obsolete (Yu et al., 2015). In order to improve the model performance by annotated training data, first the growth rate of data needs to exceed the growth rate of model parameters and second the data updates need to far exceed the emergence of new tasks, which are both quite labourious. Researchers thus generate samples to create fresh datasets, which speeds up model iteration and lowers the cost of data annotation (Yu et al., 2015; Li et al., 2021; Zhao et al., 2020; Shen et al., 2021). Also, pre-training methods that have achieved great results are used to solve this challenge (Qiu et al., 2020; Tan et al., 2018; Zaib et al., 2020; ur Rehman et al., 2019), such as BERT (Devlin et al., 2019). Despite their success and their universality, these methods still need a lot of annotated data, which bring unnecessary effort. According to studies, scientists spend over 80% of their time processing data rather than constructing models (Wu et al., 2021). HITL learning is proposed in different AI subfields and on various stages of the training loop (data preprocessing, training, evaluation) to take on this challenge, which focuses on adding human expertise into the modelling process to overcome these issues (Kumar et al., 2019).

Data augmentation by leveraging human feedback can be a great boost in the HITL pipeline (see Figure 2.20). Kirstain et al. (2021) determine that task format considerably affects the performance improvement and that from a practitioner's

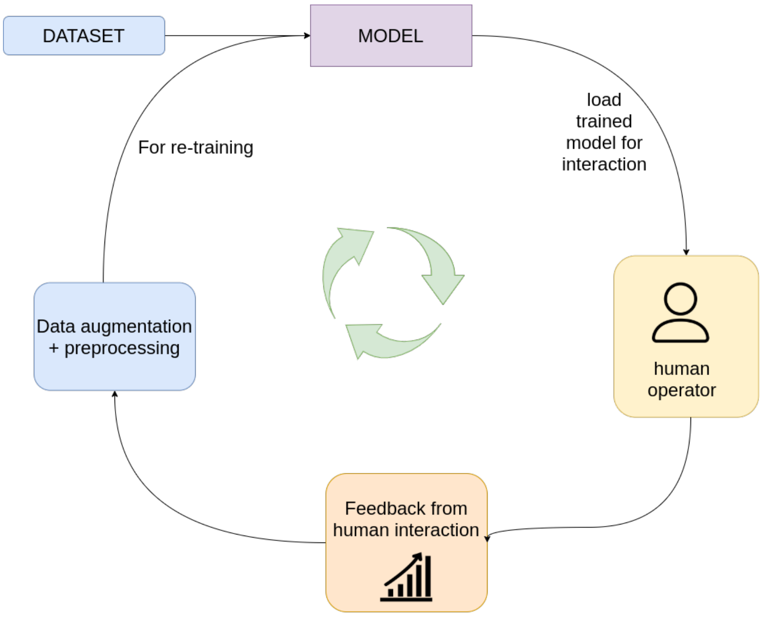

DATASET
MODEL
load trained model for interaction
For re-training
human operator
Data augmentation + preprocessing
Feedback from human interaction

Figure 2.20: The workflow of Human-in-the-loop (HITL). The human participants provide various feedback of the model performance, so as to evaluate it, and boost the performance of the model.

perspective, for many tasks where data is very sparse, the strategy of simply collecting a few hundred samples of training data will often be a more effective strategy than scaling up the model size by billions of parameters.

One way to do this is by transforming $\mathbf{D_{train}}$ by adding some variation into several samples, e.g. using handcrafted rules. On image tasks, for example, this would mean scaling, rotating or cropping images from the initial data $\mathbf{D_{train}}$ before starting the training process. Another way to achieve this augmentation is by *learnt transformations*, which is basically the same thing as handcrafted rules but this time learnt transformations were used that were trained on similar larger datasets. For instance, Kwitt et al. (2016) use a series of independent attribute strength regressors learnt from a large set of scene photos to transform each $\mathbf{x_{(i)}}$ into many new samples, then assign these new samples the label of the original $\mathbf{x_{(i)}}$. Similarly, by iteratively aligning each sample with other examples, Miller et al. (2000) learn a set of geometric transformations from a similar class. These learnt modifications are then applied to each sample $\mathbf{x_{(i)}}$, resulting in a big data set that can be learnt regularly.

Transformations do not always happen in the data set at hand. Other data sets can be transformed and adapted to be like the target $\mathbf{x_{(i)}}$, with respect to the supervised information $\mathbf{D_{train}}$. For instance, a data set of conference chairs is similar to another data set of armless chairs. Gao et al. (2018) designed a method based on Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) to generate indiscriminate synthetic $\mathbf{x}'$ aggregated from the data set of many samples.

Active learning (AL) focuses on figuring out how to label the fewest number of samples while yet achieving the greatest performance gains. In order to minimise the cost of labeling while preserving performance, it seeks to selectively choose the most useful samples from the unlabelled dataset and pass it on to the oracle (for example, a human annotator) for labeling. AL can be split in three categories: *Membership query synthesis* (Angluin, 1988), *stream-based selective sampling* (Dagan and Engelson, 1995), and *pool-based sampling* (Lewis and Gale, 1994). The learner can ask to query the label of any unlabelled sample in the input space, even the sample that the learner generated. This is known as membership query synthesis. Additionally, the main distinction between stream-based selective sampling and pool-based sampling is that the former independently determines whether each sample in the data stream needs to query the labels of unlabelled samples, whereas the latter selects the best query sample based on the assessment and ranking of the entire dataset. AL is a useful method that can be used to adapt the model to situations that were not covered in the original training samples. Elahi et al. (2016) have applied AL in Recommender Systems, where instead of just providing recommendations to the user they simultaneously learn and collect more data about the problem at hand. Qian et al. (2020) propose PARTNER a deep learning-based entity name understanding system. PARTNER is an active learning and weak supervision strategy for creating a deep learning model to recognise entity name structure with minimal human assistance. PARTNER also allows users to create intricate normalisation and variant generation functions without having to know how to code. Liu et al. (2019) propose a human-in-cycle model based on reinforcement learning (RL), which overcomes the limitations of pre-labelling and constantly improves the model through data collection. They created a deep reinforcement active learning (DRAL) approach to aid RL agents in dynamically selecting training samples by human annotators.

Researchers are also focusing on increasing and refining data on a new activity, in addition to directly applying RL for dynamic learning. Traditional facial expression recognition algorithms can only handle the seven most basic facial expressions: happy, sadness, fear, rage, disgust, surprise, and contempt. In real life, however, dealing with more micro-expressions is more vital. Butler et al. (2020) propose a micro-expression recognition method based on the HITL framework. This technique provides a customisable interface for manually proofreading automatically produced tags, ensuring the enlarged dataset's accuracy and usability. Using HITL learning, Yu et al. (2015) proposed a partially automated labelling system to save on human labour costs. They sample each subset from a huge set of candidate photos for each category iteratively, ask users to label them, classify them with the trained model, then divide the set into positive, negative, and unlabelled categories based on classification confidence.

Several HITL-related frameworks have recently been developed to apply conversation and Question Answering (QA). The goal of RQ4 is to allow conversational agents to speak with users using HITL frameworks and improve in the long run. An NLP algorithm is inserted into HITL framework that uses active learning to gather more data from experts while performing the task at hand. After the collection, these data are automatically augmented and added in the training loop of the NLP algorithm. There are two types of HITL dialogue systems: online feedback loops and offline feedback loops (Wang et al., 2021).

A huge set of human feedback must first be collected as a training set for the **offline feedback loop**, and then this training set is utilised to update the model. Wallace et al. (2019), for example, try to come up with novel adversarial cases that can fool its QA system, and then uses these examples for adversarial training.

Human feedback is used to continuously update the model in the **online feedback loop**. Researchers have shown that using online reinforcement learning to update the model with human feedback can outperform traditional approaches that mismatch the training set and online use case for dialogue systems. Hancock et al. (2019), for example, propose a lifelong learning framework. When the conversational agent's engagement with users goes well, the self-feeding mechanism in this architecture allows it to generate new examples, in order to re-train itself.

A HITL system can be created by an iterative and interactive continuous update of ground truth data but there are challenges that need to be taken into consideration: (1) the lack of trust or confidence (Smith et al., 2018); more in-depth user studies need to be designed and conducted to evaluate the effectiveness and robustness of HITL frameworks in addition to model performance (Smith et al., 2018); a paradigm to rate the quality of the collected user feedback, as user feedback can be noisy (Kreutzer et al., 2021); and since there is no effective test benchmark used by the community, the development of an evaluation method and benchmark for the HITL system.

Yu et al. (2017) concentrate on a system that learns to identify and describe visual attributes, such as shape and colour, through interaction with human tutors, incrementally, over time, trained on real human-human tutoring data. Their lifelong interactive learning period touches RQ4 but instead of HITL learning their system uses Reinforcement Learning (RL) that can handle real human conversations and reach adequate performance while minimising human effort in the learning process. Results show that the RL learnt policy achieves the aims of the task (learning visual attributes of objects) and it discovers a better balance between accuracy and learning cost than hand-crafted rule-based policies.

2.3.1 Transfer Learning in Vision and Language

To solve the data scarcity problem, related to RQ1, Transfer Learning (TL) is introduced. TL assumes training one model for one task and then re-using the knowledge it has learnt, partly or fully, on another task. It works in two stages: (1) a pre-training stage where the "base" model is trained on large general-purpose datasets; (2) a fine-tuning stage where the base model is trained on a target task, with any additional task-specific trainable parts. It was first introduced in the Computer Vision (CV) community, with large-scale pre-training on image recognition datasets (e.g. ImageNet (Deng et al., 2009)) becoming a common practice. NLP experienced a similar transformation with the appearance of word embeddings.

Word embeddings represent words as real-valued vectors in a low-dimensional trainable space, instead of having "1-hot" vocabulary-sized binary vectors of zeros, with a "one" corresponding to the word's index. Word2Vec (Mikolov et al.,

2013) and GloVe (Pennington et al., 2014) both try to form dense and expressive vector representations, trained from a given text collection of text documents. FastText (Joulin et al., 2017) is an extension of Word2Vec but unlike Word2Vec first breaks the words into n-grams, alleviating the inability of the previous model to encode unknown or out-of-vocabulary words. The aforementioned models, despite their high performance in various NLP tasks, are not able to take word context into consideration.

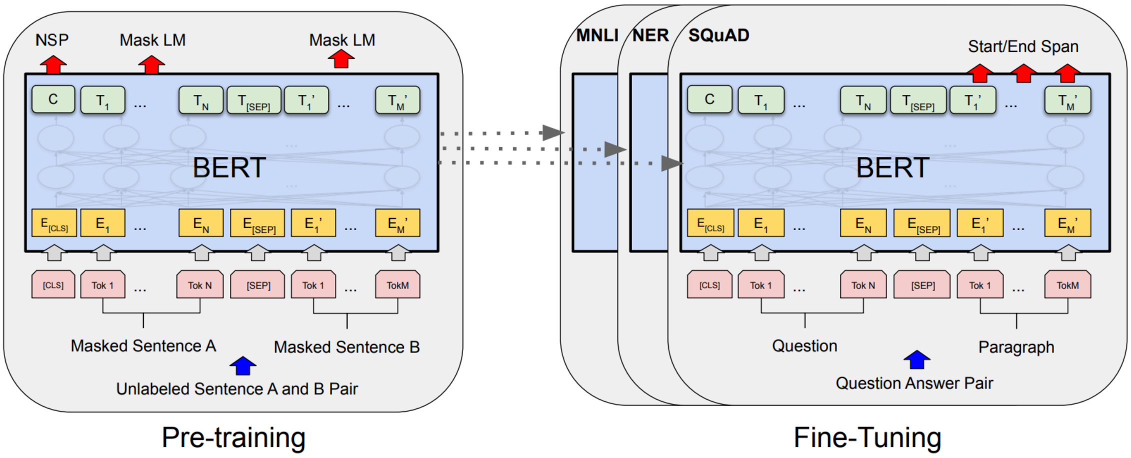


Figure 2.21: Overall pre-training and fine-tuning for BERT. Apart from output task-specific layers, the same architectures are used in both pre-training and fine-tuning. (Devlin et al., 2019)

Contextual Word Embeddings based on LSTMs (Hochreiter and Schmidhuber, 1997) and Transformers (Vaswani et al., 2017b) were introduced to solve this issue. Models such as ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019) provided what is referred to as a "contextual" representation, instead of being static as the predecessors. BERT, being the most used and highest performing, was designed for use in downstream tasks in a "pretrain-finetune" fashion (see Figure 2.21). Before applying BERT on a task, we first organise the input in two pairs, i.e context/response pairs in a downstream task, and second pre-train using usually two objectives: (1) Language Modelling (LM); (2) Next Sentence Prediction (NSP).

More recently, Transformer-based GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) brought transfer learning for dialogue (Wolf et al., 2019; Budzianowski and Vulić, 2019). These transformer-based models use a self-attention mechanism, i.e. BERT, and consist of billions of parameters and are pre-trained on large text corpora, consisting of millions of documents. Their core head can be fine-tuned and used for gradient-based task. Models like ELMo, BERT, GPT-2 and GPT-3 showed that transfer learning can be applied in NLP as well as in vision and is now considered among the best practices to approach new NLP problems via a "pretrain-finetune" fashion instead of training from scratch. BERT embeddings are used in various experiments across this thesis.

2.4 Discussion

In conclusion, language grounding refers to the interpretation of language in a situated context. Earlier research (Chai et al., 2016; Boye et al., 2014; Blukis et al., 2018; Misra et al., 2018) has focused on directly mapping natural language commands to actions on a 3D virtual world instead of a 2D map or chart. Furthermore, the perspective of prior work is mostly first person, such as the TeaCH dataset (Padmakumar et al., 2021), where despite the fact that conversational dialogues were used to perform object interaction, such as picking up an object, and state changes, such as slicing bread, it is not high risk and is looking to ground elements from an image instead of entities on the map (Götze and Boye, 2016). All of the aforementioned works, neither investigate the problem of grounding language to maps and charts, nor learn to read maps so as to work in unison with human experts. This thesis focuses on human-robot collaboration, where a CA acts as a mediator between the human operators and the UXV, with 2D maps or charts acting as the real world of interaction.

This chapter provided an introduction to conversational agents and situational discourse. It has served as a resource for the community's literature review as well as background reading of SOTA. It included an overview of how the research community has approached knowledge of situational instruction and navigation tasks. To acquire a better view on how spatial challenges in dynamic environments, such as highly structured maps and GPS location data, are addressed, common datasets and methodologies used in vision and language grounding are discussed. Finally, an introduction to Human In The Loop (HITL) learning and how it is being utilised by the community to train low-resource data is offered, as well as a basic overview of the usage of transformers in vision-language tasks. We will refer to the aforementioned work throughout the thesis and compare to the methods presented here.

Chapter 3

A Wizard Of Oz Study on Interactive AUV Planning

The procedure for gathering the right requirements when designing a system for an expert domain can be rather complex. Even the experts themselves find it hard to specify what tasks they might be carrying out. Furthermore, it may be unwise to make assumptions about what users would want to do in a new domain and how they will want to do it a priori. Methods for simultaneous prototyping/design, requirements collecting, and evaluation should be employed to address these issues. The Wizard of Oz (WOz) methodology has been used to develop and evaluate natural language interfaces (Lyons et al., 2005; Rajman et al., 2006; Marge et al., 2017). WOz serves as an ideal dialogue system for a new task at hand, providing insights on how human subjects would interact with such a system, assisting in selecting the right research topics that need to be addressed, and providing data for new tasks, especially in low-resource domains (Arslan and Eryiğit, 2021; Weng et al., 2006). In a WOz method, a human "wizard" mimics aspects of an interactive system that has not been implemented yet, with the goal of determining the interaction model for a specific application. The information gathered will be utilised to create language models and dialogue tactics for the system. Furthermore, this enables both the online analysis of confusing input and the collection of real data, without requiring a significant investment in implementation.

The ultimate purpose of this thesis is to create a model for instructing RAS systems that optimises interaction for plan quality and speed, thus tying interaction style to extrinsic task success metrics. This chapter presents a Wizard of Oz (WOz) study to gather data and investigate the way human operators would collaboratively make plans via a conversational "planning assistant" for remote autonomous systems (RQ1). As autonomous systems become more commonplace, we need a way to easily and naturally communicate to them our goals and collaboratively come up with a plan on how to achieve these goals.

In this chapter, we focus on the emergency response domain of Autonomous Underwater Vehicles (AUV) and our goal is to study the AUV mission planning process using multi-modal interaction by creating a wizard-controlled collaborative multi-modal planning system in the form of a conversational agent, called VERSO.

The WOz study is described here and analysed as it was exhibited in the Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP) at the North American Chapter of the Association for Computational Linguistics (NAACL) 2019 (Katsakioris et al., 2019). The contributions include:

1. A dialogue framework and the implemented software to conduct multiple-wizard WOz experiments for multi-modal collaborative planning interaction;
2. A corpus of 22 dialogues on 2 missions with varying complexities;
3. A corpus analysis indicating that incorporating an extra modality in conjunction with spatial referencing in a chatting interface is crucial for successfully planning missions.

The chapter consists of three parts: Section 3.1 presents our WOz setup, participants and methodology of the experiment; in Section 3.2 we analyse the collected corpus with subjective and objective measures; Section 3.3 discusses observations on the instructions used at this corpus and possible future work, and summarises the chapter.

3.1 Method and Experiment Set-up

Our 'planning assistant' conversational agent will interface with planning software called SeeTrack provided by our industrial partner SeeByte Ltd. SeeTrack can run with a real AUV running SeeByte's Neptune autonomy software or in simulation.

This software allows the planning of missions by defining a set of objectives with techniques described in (Lane et al., 2013a; Miguelanez et al., 2011; Petillot et al., 2009). These can include, for example, searching for unexploded mines by surveying areas in a search pattern, while collecting sensor data and if, for example, a suspect mine is found then the system can investigate a certain point further (referred to as target reacquisition). Similarly to (Bonial et al., 2017), we used two wizards for our experiment, as piloting showed the workload too great for one Wizard. See Figure 3.1 for the set-up. We refer here to the wizards as 1) Chatting-Wizard (CW), who alone communicates with the subject getting information that is required to create the plan; and 2) the SeeTrack Wizard (SW) who sits next to the CW and implements the subject's requirements into a plan using SeeTrack and passes plan updates in the form of images to the CW to pass onto the subject. The subject was in a separate room from the wizards and interacted via a chat window for receiving and sending texts and receiving images of the updated plan.

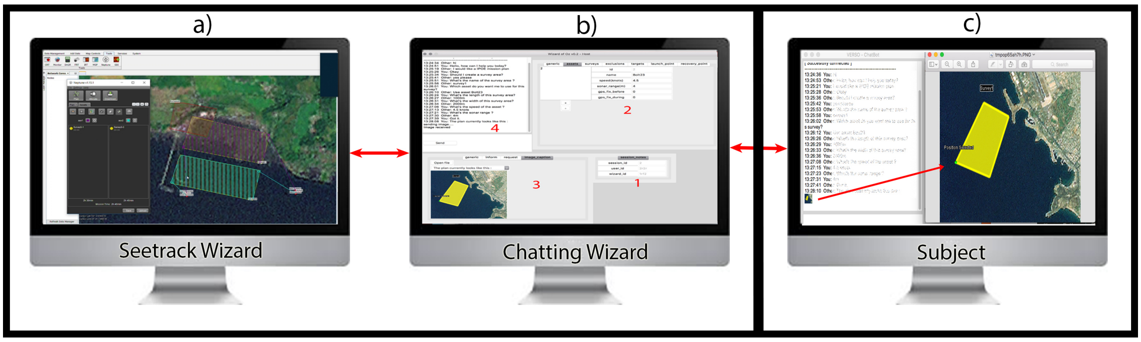
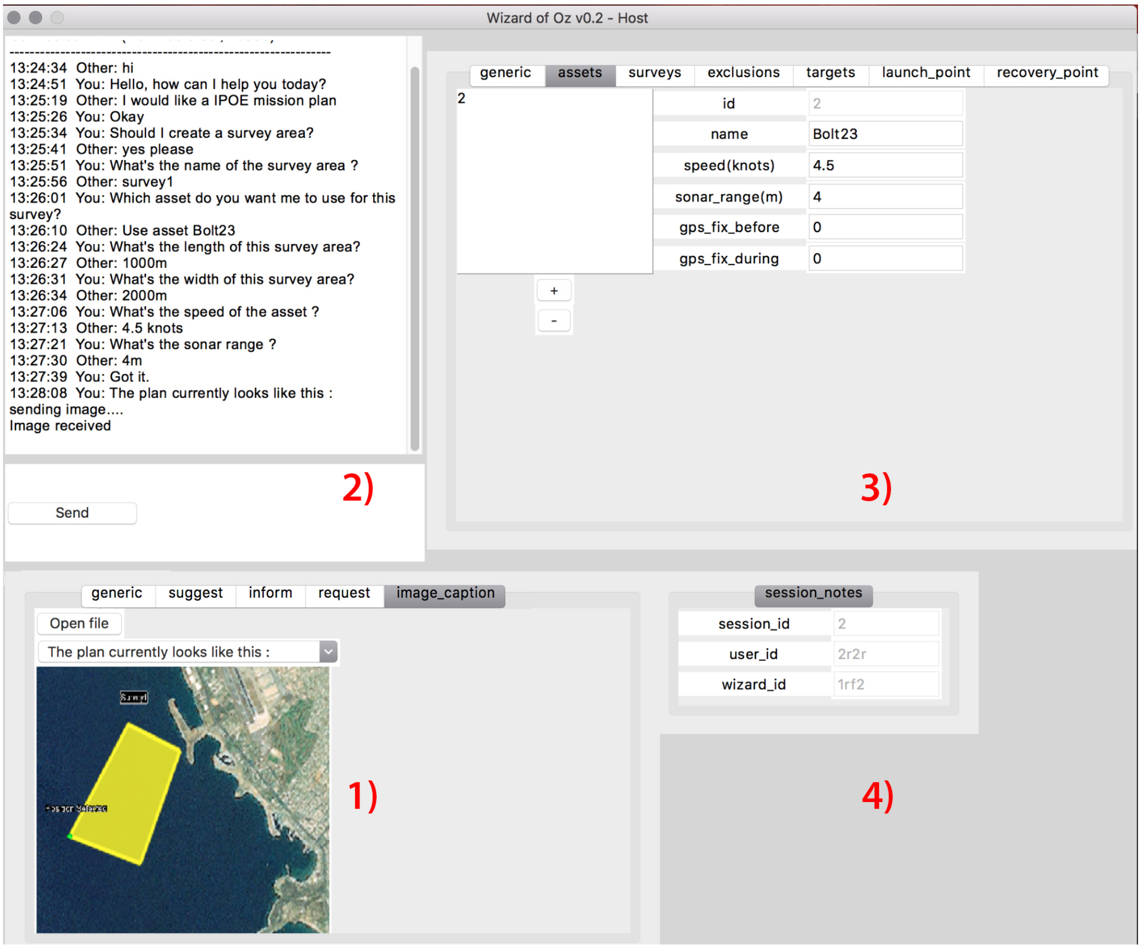


Figure 3.1: Experimental Set-Up, where a) SeeTrack Wizard, b) Chatting Wizard and c) Subject.

Before designing the WOz experiment and software, we had to understand the language and the complexities of our domain. To establish the main actions and dialogue act types for the system to perform, we recorded an expert planning a mission on SeeTrack whilst verbalising the planning process and his reasoning. Similar human-provided rationalisation has been used to generate explanations of deep neural models for game play (Ehsan et al., 2017). After analysing the expert video, we implemented a multi-modal Wizard of Oz interface that is capable of sending messages, either in structured or free form, and images of the plan. The GUI is made up of four windows (see Figure 3.2). The first has all the possible dialogue acts (DA) the wizard can use together with predefined utterances for expedited re-



```
-----------------------------------------------------------------------
13:24:34  Other: hi
13:24:51  You: Hello, how can I help you today?
13:25:19  Other: I would like a IPOE mission plan
13:25:26  You: Okay
13:25:34  You: Should I create a survey area?
13:25:41  Other: yes please
13:25:51  You: What's the name of the survey area ?
13:25:56  Other: survey1
13:26:01  You: Which asset do you want me to use for this
survey?
13:26:10  Other: Use asset Bolt23
13:26:24  You: What's the length of this survey area?
13:26:27  Other: 1000m
13:26:31  You: What's the width of this survey area?
13:26:34  Other: 2000m
13:27:06  You: What's the speed of the asset ?
13:27:13  Other: 4.5 knots
13:27:21  You: What's the sonar range ?
13:27:30  Other: 4m
13:27:39  You: Got it.
13:28:08  You: The plan currently looks like this :
sending image....
Image received
```

2

id	2
name	Bolt23
speed(knots)	4.5
sonar_range(m)	4
gps_fix_before	0
gps_fix_during	0

The plan currently looks like this :

session_id	2
user_id	2r2r
wizard_id	1rf2

Figure 3.2: Wizard of Oz interface (Figure 3.1b) used by the Chatting Wizard, 1) Dialogue acts and structured prompts, 2) Chatting window, 3) State of the plan in the form of value-slots, 4) Session notes.

sponses. Once a DA is selected, the pre-defined text appears in the chat window, from there the CW is able to modify as needed. The third window allows the CW to insert values (also referred to as 'slots') needed for the plan obtained through interaction from the user. Finally, the fourth window is for recording session details such as subject ID. The CW works collaboratively with the subject to develop a list of the necessary parameters that the SW needs to create the plan.

Each subject was given 1) a short questionnaire to collect demographic information, 2) the Object-Spatial Imagery and Verbal Questionnaire (OSIVQ) (Blazhenkova and Kozhevnikov, 2009) questionnaire discussed below and 3) instructions on how to approach the task of planning a mission using a conversational agent. They were then given instructions for two missions and told that they had to plan them both with a time limit of one hour. The wording of the mission was very high level, so as not to prime the subjects. There are many elements that make up a plan (e.g.

survey areas, exclusion zones, targets to investigate) and, therefore, many variations are possible. As we did not want to bias them and we wanted a variety of plans, we gave them no instruction as to what elements to include in the plan. Once both missions were completed a post-questionnaire was administered to obtain their subjective opinion on the planning process. Appropriate consent was obtained and all experiments going forward were approved by the HWU ethics committee. Subjects took part on a voluntary basis, with the time spent on participation being part of their professional work that day.

3.1.1 Subject Group

Planning missions for AUVs is a complex task, especially in the case of sophisticated software, such as SeeTrack. For this reason, we decided to focus our study on expert users who are familiar with SeeTrack and AUVs. We recruited subject human operators from our industrial partner SeeByte Ltd, who all had some experience with SeeTrack but none of them had collaboratively created a plan before either by chatting with a human or a chatbot. There were 11 experts (10 male and 1 female), exactly reflecting current gender proportions of employees in the engineering and technology sector in the UK (9% female (Technology, 2015)). Further demographics showed that the majority of them were aged between 25 - 35, educated to undergraduate, masters, PhD degree level, with roles such as development and software engineers. Half of our subjects were native English speakers.

We used the validated Object-Spatial Imagery and Verbal Questionnaire (OSIVQ) (Blazhenkova and Kozhevnikov, 2009), in order to take into account the visual and verbal cognitive styles of our subjects (see Figure 3.3). Each OSIVQ item is a 5-point Likert scale item, with 45 items forming three subscales. Subjects completed the OSIVQ following its standard instructions with their responses collated into three subscale scores (OSIVQ Object, Spatial and Verbal). Generally our expert subjects scored higher on the spatial cognitive style, which is perhaps not surprising given that they came from a pool of software developers.

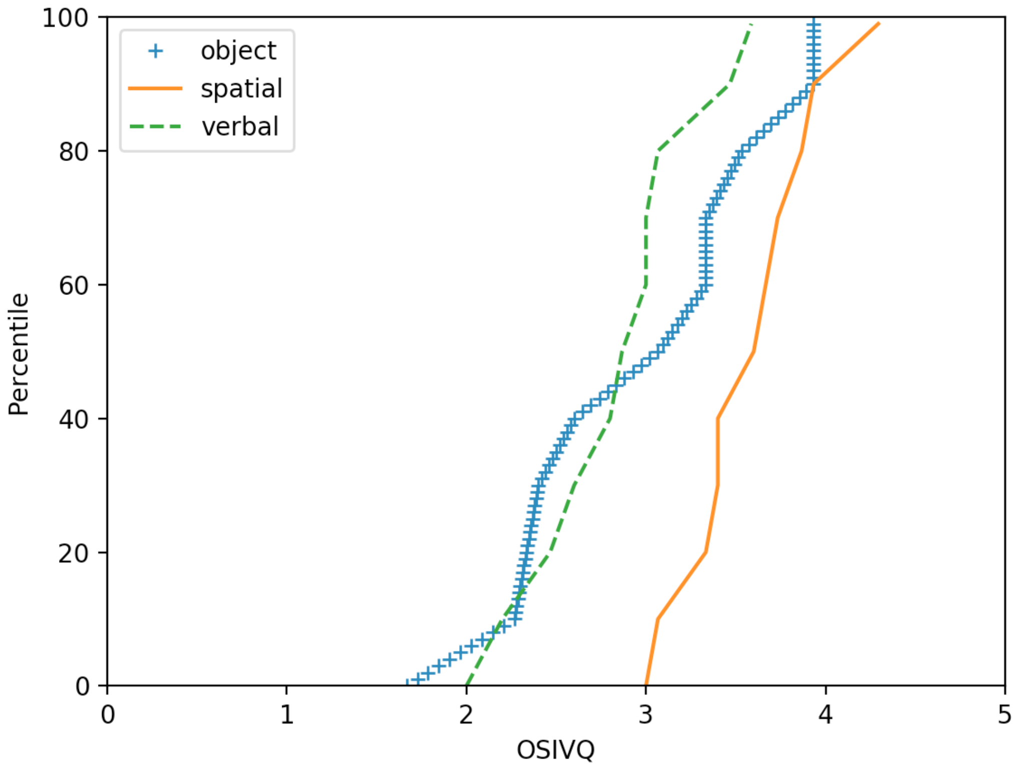


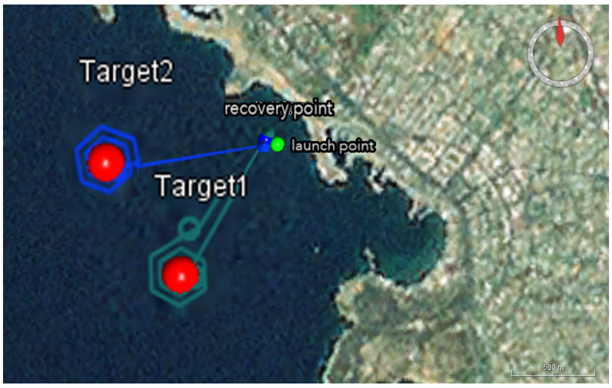
Figure 3.3: The correspondence of raw OSIVQ 5-point Likert scale scores to percentiles. The distribution of the raw scores on the three scales of OSIVQ in terms of percentiles (Blazhenkova and Kozhevnikov, 2009).

3.2 Corpus Analysis

We collected 22 dialogues between the wizards and the subjects (2 missions per subject), which due to their small size were analysed by a single annotator (the author) and checked by an expert (industrial supervisor). Due to the size of the dataset no further annotators were employed. Figure 3.4 shows an example of a dialogue interaction with their corresponding dialogue acts. We split our analysis into Objective and Subjective measures.

3.2.1 Objective Measures

In order to capture a wide variety of data and to measure the difficulty of the task, the first mission was, by design, harder with more complicated objects to describe, and thus took the subjects longer to plan in terms of actual planning time and number of user/system turns (see Table 3.1).



1. USER: 'move t2 200m west of t1'[inform]
2. SYSTEM:'Could you repeat that in different words? t1? t2?'[repeat]
3. USER:'move target2 200m west of target1'[inform](spatial reference to target1)
4. SYSTEM:'Okay'[ack]
5. SYSTEM:'Let's use asset1 for target1'[suggest]
6. USER:'sounds good'[affirm]

Figure 3.4: Dialogue excerpt and the corresponding image provided by the system, displaying 2 targets, a launch point and a recovery point; the user is trying to move "target2" west of "target1" (the SYSTEM was controlled by a wizard).

We used the ISO (24617-2:2012) standard as a starting point for dialogue act annotation and devised the Dialogue Acts (DAs) below. DAs are proposed as annotated data in order to train NLP algorithms in the future. In addition, analysis of the DAs showed the preferred actions by operators when planning missions with a CA and for isolating the templated instructions in Section 4.2.3. Figure 3.5 gives the distribution of dialogue acts, which were categorised into five groups:

1. **Generic**: conversational acts such as greetings, acknowledgements;
2. **Inform**: informing of values for slots such as the required position of the objects on the map;
3. **Request**: requesting information such as current values of slots;
4. **Suggest**: for making suggestion e.g. positions of entities;
5. **Image**: for the user requesting and for the system sending plan images.

The most frequent user DA is the "inform" dialogue act (54%), which informs the system about the plan slot values. This dialogue act is also used for utterances that instruct the system to move objects around on the map by referring either to the object's position or to nearby objects. 53% of these "inform" acts contain referring

Measures	Mission 1	Mission 2
# of turns	26.4(9.1)	13.1(4.4)
# of system turns	51.4(21.0)	27.0(7.5)
# of user turns	36.4(14.4)	19.7(8.1)
# of produced images	8.8(3.8)	5.0(1.3)
Time-on-Task (min)	26.3(0.005)	14.5(0.004)

Table 3.1: Measures per dialogue for each mission [mean(sd)].

expressions (see lines 1 and 3 of Figure 3.4 for examples). In addition, it is clear that, due to the spatial nature of the tasks, the extra modality of plan images is key to success, as reflected by the frequency of Image dialogue acts (around 16% of the total dialogue acts). These DAs include the user requesting a plan image 'show_picture' or 'image_caption' where the system, either proactively or as a response to a user request, sends an image of the plan. The most used DA by the wizard was "ack" 30%, used for acknowledging information (e.g. "okay","got it" etc.).

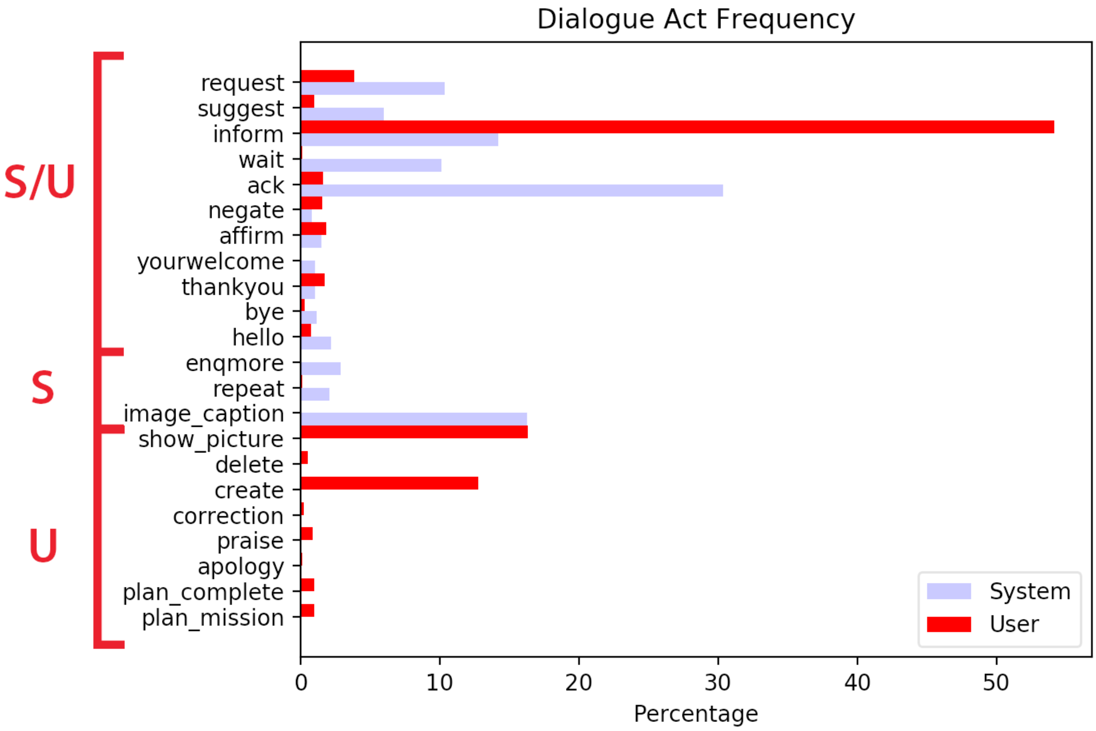


Figure 3.5: Dialogue Act Frequency of the 22 DA types. "S" is for system only DA, "U" is for user only DA and "S/U" refers to DA that both the system and the user used.

3.2.2 Subjective Measures

The subjective quality of the plans was measured by an expert using a 5-point Likert scale (see Table 3.2). At least 45% of the plans for both missions were measured "High Quality", with a greater number of lower-rated plans for Mission 1. No correlation was found between time-on-task and quality of plans, however, subjective feedback indicates that subjects would have liked more time to improve their plans. The response time of the Wizard was slow (average 15sec), which is a typical issue in WOz studies and we believe that an implemented system would allow for a quality plan to be generated within a reasonable amount of time. Nevertheless, varying quality of plans results in a rich dataset, that will enable the system to learn better strategies for creating optimal plans as well as coping strategies. There is a medium-strong positive correlation of $r = 0.59$ (Spearman's Correlation) between the expertise of the subjects (as determined by the pre-questionnaire) and the quality of the plan for the first mission indicating that, perhaps unsurprisingly, experts were able to produce plans with better quality.

The post-task questionnaire measured the subjective scores for User Satisfaction, the pace of the experiment and the importance of multi-modality. Specifically, the following questions were asked on a 5-point Likert scale:

Quality	Mission 1	Mission 2
Very High Quality	0%	9%
High Quality	45%	45%
Neutral	9%	36%
Low Quality	18%	9%
Very Low Quality	27%	0%

Table 3.2: The quality of all 22 plans measured by an expert using a Likert scale

Q1: I felt that VERSO understood me well

Q2: I felt VERSO was easy to understand

Q3: I knew what I could say at each point in the interaction

Q4: I felt that the pace of interaction of VERSO was appropriate

Q5: VERSO behaved as expected

Q6: It was easy to create a plan with VERSO

Q7: From my current experience with using VERSO, I would use the system regularly to create plans

Q8: The system was sluggish and slow to respond (reversed)

Q9: The screen shots of the plan were useful

Q10: The screen shots of the plan were sent frequently enough

Mean User Satisfaction is 3.5 out of 5, calculated as an average of Q1-7, which are questions adapted from the PARADISE evaluation framework (Walker et al., 1997). Q8 reflects the speed of the interaction with the mean/mode/median as 4/4/4. This score is reversed and so these high scores indicate high perceived slowness. As mentioned above, this is a common problem with wizarding set-ups. Q9 and Q10 refer to the images sent and we can see from the mean/mode/median of 4.6/5/5 for Q9 that images were clearly useful but, perhaps could be sent more frequently (3/4/3 for Q10). The users' preference for images of plans may be related to their cognitive styles being mostly spatial, as mentioned above.

Figure 3.6 gives perceived workload collected after both tasks by NASA TLX (Dickinson et al., 1993), with low scores indicating low workload. The overall mean Raw TLX (RTLX) score of 46 is comparable to a study for remote controlling robots through an interface as reported in (Kiselev and Loutfi, 2012). However, a comparative study would be needed with SeeTrack to understand the costs/benefits in terms of cognitive workload of a conversational assistant vs. a visual interface.

3.2.3 Qualitative Feedback

Subjects were asked 2 open questions: 1) *Tell us what you liked about VERSO* and 2) *Tell us what you didn't like about VERSO*. An inductive, thematic analysis was done inspired by an open thematic coding (Corbin and Strauss, 2008; Strauss, 1987). Themes identified include:

Theme 1: Suggestions for extra functionality: Some subjects were not sure if the program crashed or if it was working in the background. We had a dialogue act "wait" but feedback indicated it would be better to have a visual indicator as well. Note, in the actual future working system, we will not have the same delays as in the WOz experiment.

Theme 2: Map meta-data: Some subjects (P5 most specifically) desired more meta-data on the plan images they were receiving when referring to an object. e.g. *"not always clear where the frame of reference is"*. When performing spatial tasks

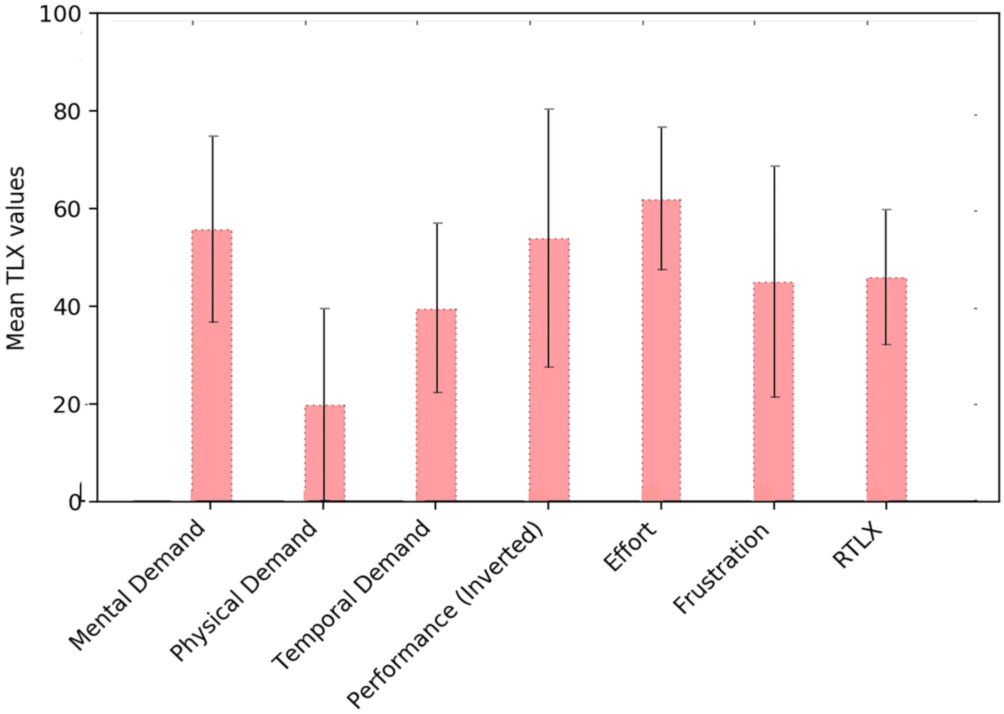


Figure 3.6: Mean Raw TLX scores, with high values indicating high workload. Error bars indicate standard deviation.

on a map, clear referring points are crucial and meta-data on the map, such as entity names (as with the Map Task (Anderson et al., 1991) landmarks), would help establish grounded referring expressions between participants.

Theme 3: Mixed initiative & Handling multiple requests: The WOz interface was designed as a mixed-initiative dialogue system, capable of suggesting actions and the subjects seem to like this type of interaction e.g. *P1:"being prompted to add information that could be forgotten."* Also noted was the 'system's' ability to handle multiple requests in a single utterance.

User Utterance	Reference
plan a survey area 200 meters south ***of the initial point.***	**of the initial point**
Create an exclusion zone ***along the shoreline.***	**along the shoreline**
create another exclusion zone east ***of the survey area.***	**of the survey area**
Does Exclusion0 contain ***the shore?***	**the shore**
create a launch and recovery point 500m south ***of target0.***	**of target0**
make launch point 100 m ***from the shore.***	**from the shore**
yes, south ***of glyfada beach.***	**of glyfada beach**

Table 3.3: Example utterances from our WOz study, that included references to landmarks and other objects.

3.3 Discussion

Operators tended to use landmarks available on the chart, such as “shore”, “coastline”, “beach”, as references to move and manipulate objects (see Table 3.3). Whilst some were keen on using only movable objects, such as “targets”, “surveys” as references as well. We thus need an accurate way of understanding referring expressions and interpreting them in terms of geographical coordinates.

This chapter presents a two-wizard WOz study for collecting data on a collaborative task, identifying the importance of mixed modalities and object referencing, for successful interaction during mission planning.

In Section 4.2, we use templates taken from the WOz study described in this chapter to generate a synthetic dataset, so as to train neural models, as described in Chapters 6 and 7. However, such generated data do not always reflect the true distribution of human natural language. Therefore, further data collection on Amazon Mechanical Turk using Open Street Maps was conducted and described in Chapter 4, in order to reach a wider subject pool and compensate for the gender imbalance.

Chapter 4

Crowdsourced Data Collection for a Low-resource Expert Domain

This chapter builds on the WOz study from Chapter 3, where the Wizard played the role of an intelligent agent to assist human operators with planning missions for Autonomous Underwater Vehicles (AUV). The data collection that is presented in this chapter is described and analysed as it was exhibited in the International Conference on Multi-modal Interaction (ICMI) 2020 (Katsakioris et al., 2020).

The chapter consists of two parts: (1) we present the publicly-available Robot Open Street Map Instructions (ROSMI) corpus, a rich multi-modal dataset of map and natural language instruction pairs that was collected via crowdsourcing to aid the advancement of reference resolution and robot-instruction understanding (RQ2) (Section 4.1); (2) Section 4.2 introduces the Robot Electronic Navigational Chart Instructions (RENCI), a synthetically generated dataset, similar to ROSMI but based on Electronic Navigational Charts (ENCs), from human examples collected during the WOz study in Chapter 3. Finally, Section 4.3 summarises the chapter.

4.1 Robot Open Street Map Instructions (ROSMI)

In order to capture a wider range of the human language, more data was gathered with subjects (crowd-workers), using the online platform on Amazon Mechanical Turk (AMTurk)[1]. We present ROSMI, a multi-modal corpus of visual and natural

[1] www.mturk.com

USER: Send one drone 89m south west of Chevron to put out the fire.

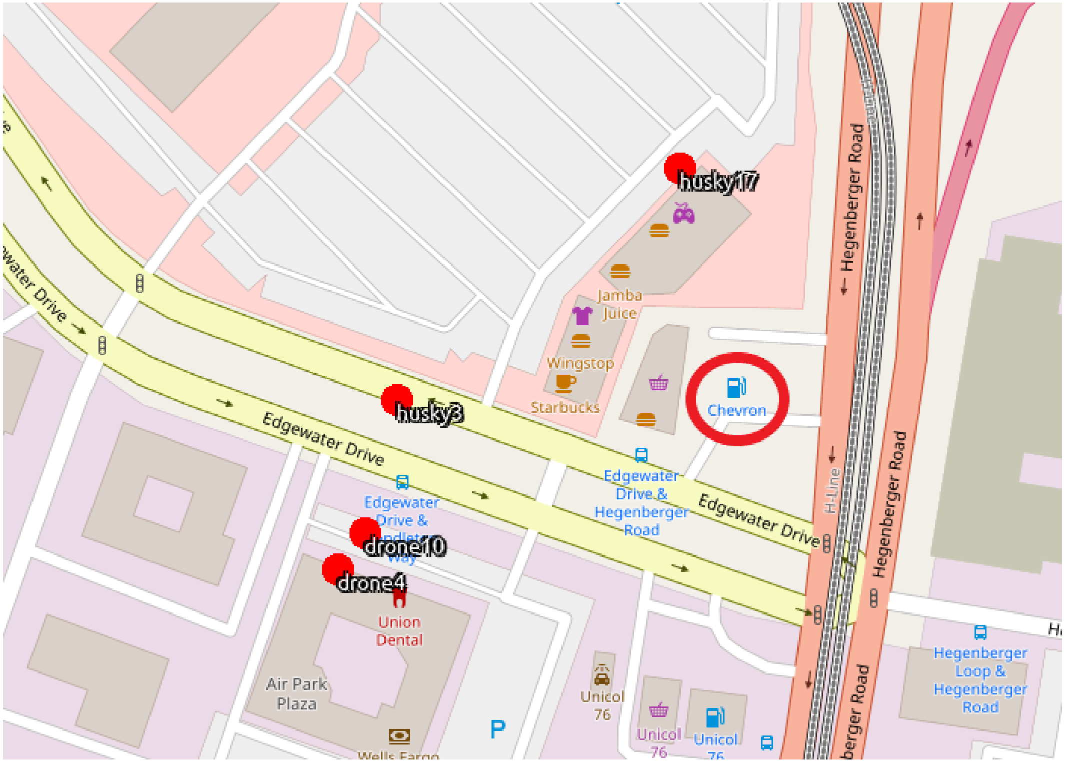
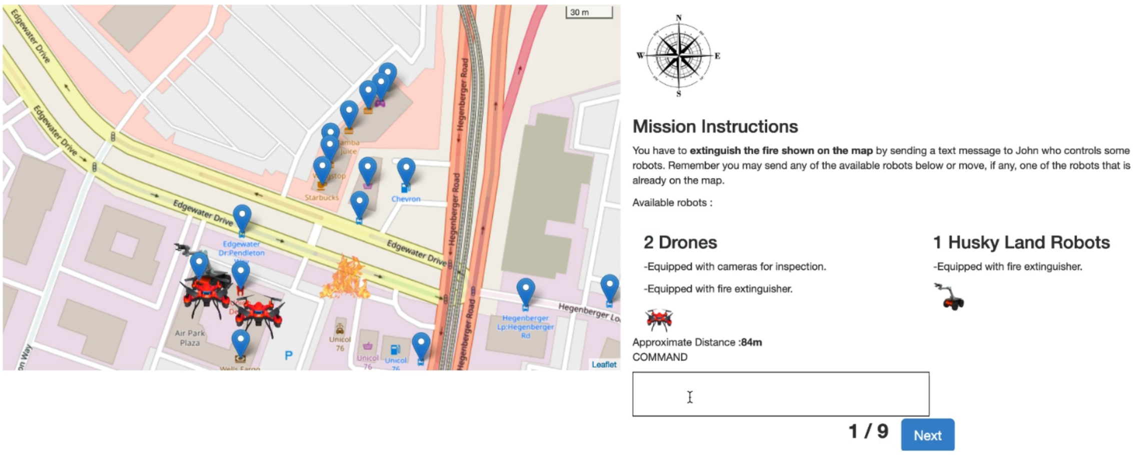


Figure 4.1: User instruction and the corresponding image, displaying 4 robots and landmarks. A circle around the target landmark has been added for clarity; subjects were not given any such visual hints.

language instruction pairs, in the domain of emergency response, whereby the subjects are given a scene in the form of a map and are tasked to write an instruction to command a conversational assistant to direct a number of robots and autonomous systems to either inspect an area or extinguish a fire. Figure 4.1 shows an example of such a written instruction. These types of emergency scenarios usually have a central hub for operators to observe and command humans and RAS to perform specific functions aided by *'Command and Control'* (C2) style interfaces, where the robotic assets are visually observable as an overlay on top of the map. These map-based interfaces are thus dynamic in nature, adding an extra layer of complexity that we attempt to capture herein. Although our focus is on emergency response resolution with robots, the data and collection method are applicable to collaborative tasks that involve moving objects with respect to a 2D map, such as in gaming and human instruction-giving.

We created an interface for data collection (see Figure 4.2). As our target audience was the general public, we formulated our task a bit differently for this data



Mission Instructions

You have to **extinguish the fire shown on the map** by sending a text message to John who controls some robots. Remember you may send any of the available robots below or move, if any, one of the robots that is already on the map.

Available robots :

2 Drones

-Equipped with cameras for inspection.

-Equipped with fire extinguisher.

1 Husky Land Robots

-Equipped with fire extinguisher.

Approximate Distance :**84m**

COMMAND

1 / 9

Figure 4.2: AMTurk interface used for the data collection. Each user was given 9 randomly generated scenarios and for each they had to give the right command accordingly.

collection. To display the missions, we use OpenStreetMap (Haklay and Weber, 2008), which are open source and are like everyday maps that we use to navigate through cities. In order to make the problem a bit more challenging, instead of only underwater vehicles, we added aerial and ground autonomous vehicles that the users could control with natural language. Rich structured data, such as plans of buildings or detailed maps, are key for successful deployment of RAS in these scenarios. However, such data is not typically easily digestible to human operators and thus interaction using natural language instructions can be limited, which can be particularly problematic in fast-moving collaborative scenarios such as emergency response.

In Chapter 3, from the set of 22 preliminary dialogues that we collected and analysed, we observed that the subjects preferred the usage of landmarks when instructing the conversational agent. Through this pilot, it was observed that there was a white space in terms of understanding instructions in a complex dynamic environment and a need to address the challenges of conversing and planning with an operator given dynamic map-based stimuli. To this end, we conducted a larger dataset collection that would have not been feasible to scale just by employing maritime experts and so, we used OSM, which is easily accessible to a layperson, while still transferable, in principle, to the nautical domain.

Our contributions are summarised as follows: 1) a novel multi-modal dataset[2]

[2] `https://github.com/marioskatsak/rosmi-dataset`

comprising of map-based instructions for human-robot interaction; and 2) the accompanying structured metadata that can be used to train language models (see Chapters 5 and 6).

As mentioned above, the task is based on OpenStreetMap (OSM) (Haklay and Weber, 2008). In the next subsection, we explore OSM in more detail.

4.1.1 OpenStreetMap

OpenStreetMap (OSM) (Haklay and Weber, 2008) is a massively collaborative project, started in 2004, with the main goal to create a free editable map of the world. The data is available under the Open Data Commons Open Database Licence (ODbL) and has been broadly used for prior research (Götze and Boye, 2016; Hentschel and Wagner, 2010; Haklay and Weber, 2008). It is a collection of publicly available geodata that are constantly being updated. It consists of many layers in order to represent different geographic attributes of the world. Physical features such as roads or buildings are represented using tags (metadata) that are attached to its basic data structures (its nodes, ways, and relations). There are two types of objects, *nodes* and *ways*, with unique IDs that are described by their latitude/longitude (lat/lon) coordinates. Nodes are single points (e.g. coffee shops) whereas ways can be more complex structures, such as polygons or lines (e.g. streets and rivers). A comprehensive list of all the possible features available as metadata can be found online[3].

4.1.2 Data Collection Method

We conducted a web-based data collection by crowdsourcing our task on AMTurk. Our goal was to collect textual commands for emergency response and study how users would refer to objects on a map in order to guide robots accurately. Each job consisted of 9 sub-tasks; each sub-task consisted of looking at a map and writing instructions for a set of robots to complete either an inspection or resolve an emergency, such as a fire. For each of the sub-tasks, a scenario was generated based on one of seven unique maps. Some maps had ocean coverage and their associated task was limited to just inspection.

[3] wiki.openstreetmap.org/wiki/Map_Features

We used three types of robots: a) Autonomous Underwater Vehicles (AUVs), b) Unmanned Ground Vehicles (UGVs/Husky) (Robotics, 2011) and c) Unmanned Aerial Vehicles (UAVs/drones). For each mission, every robot could perform either or both tasks of inspecting an area or extinguishing a fire.

Task instructions: Subjects were told that there is an "assistant conversational agent" (John), to whom they are to send instructions to direct the robots. We introduced the scenarios (fire emergency vs inspection), the features of the robots (see Figure 4.2) and told them to pick one or more robots and move them to the area of the emergency (the Emergency Zone). Each scenario has a unique target GPS location, hence one task for this data could be GPS location prediction. All nine scenarios had to be completed in 20 minutes. A video tutorial[4] was also provided, describing the task.

The detailed task description is as follows:

You are in charge of some robots to either inspect an area for fire or put out a fire. For each task you will be given a list of available robots and you may choose one or more to fulfil your mission. You can also use robots that are already on the map by referring to their pop-up names. Robots include underwater robots, husky land robots and drones. For each mission the abilities of each robot will be given as a reminder.

There are two types of tasks:

- *Inspection: you can send the robots to inspect an area indicated by a red box on the map.*
- *Fire: If you are instructed to extinguish a fire, you will see a symbol for fire on the map.*

You have an Intelligent Chatbot, as an assistant, to help you. The chatbot understands the English language well and you can text him, or in some cases, use predefined commands to send robots to either the inspection area or the fire. You may refer to any available landmarks on the map. The chatbot has the same map. Try to use direction and distance with the help of the compass and the scale. Examples:

1. *Move husky1 30m south from the Main Tower, to prevent the fire from spreading.*

[4] `https://youtu.be/dQsgBr0eg20`

2. *Send 2 drones 450m west north of the "Fiji" bus stop next to the Haydar road.*

The quality of tasks was controlled by (1) accepting only AMTurkers with a previous task success rate of more than *80%*; (2) checking the *length of the instruction*; and (3) asking AMTurkers to respond first to one *"Preliminary scenario"* correctly, before proceeding to complete the job.

In total, 127 crowdworkers completed the experiment successfully. The majority of these workers came from the US (57%) or India (32%) and more than half of the AMTurkers had some sort of college or bachelor's degrees (Ross et al., 2010)[5].

4.1.3 Corpus Description

The ROSMI corpus comprises: (1) 783 tuples of instructions, maps with metadata and target GPS locations; (2) additional manual annotations related to grounding (see below); and (3) the corresponding scenarios given to human subjects.

Each instruction datapoint was manually checked as follows: an annotator (an author), given just the natural language instruction and the map with the robots, attempted to process the instruction and identify the GPS location of the Emergency Zone. If this did not match the 'gold standard' GPS coordinates per the scenario map (i.e. where the inspection area was) or if the instructions were nonsensical then the instructions were deemed of too poor quality to be of use and were discarded.

4.1.4 Manual Annotation

The corpus was further manually annotated with: (1) a link between the NL instruction and the referenced OSM entities (referred to here as "landmark grounding"); (2) any spatial references related to the OSM entities (e.g. most northern point of the polygon); and (3) the distance and bearing from the Emergency Zone, as referred to in the NL instruction (referred to here as "entity recognition").

These are discussed in detail below.

1. **Landmark Grounding:** Each scenario has 3-5 generated *robots* and an average of 30 *landmarks* taken from OSM. Each subject could refer to any of these objects on the map, in order to complete the task. Grounding the correct noun phrase to the right OSM object or robot is crucial for predicting accurately

[5] Due to the anonymous nature of AMTurk, we cannot report exact gender statistics.

USER: Send 2 drones near Harborside Park.

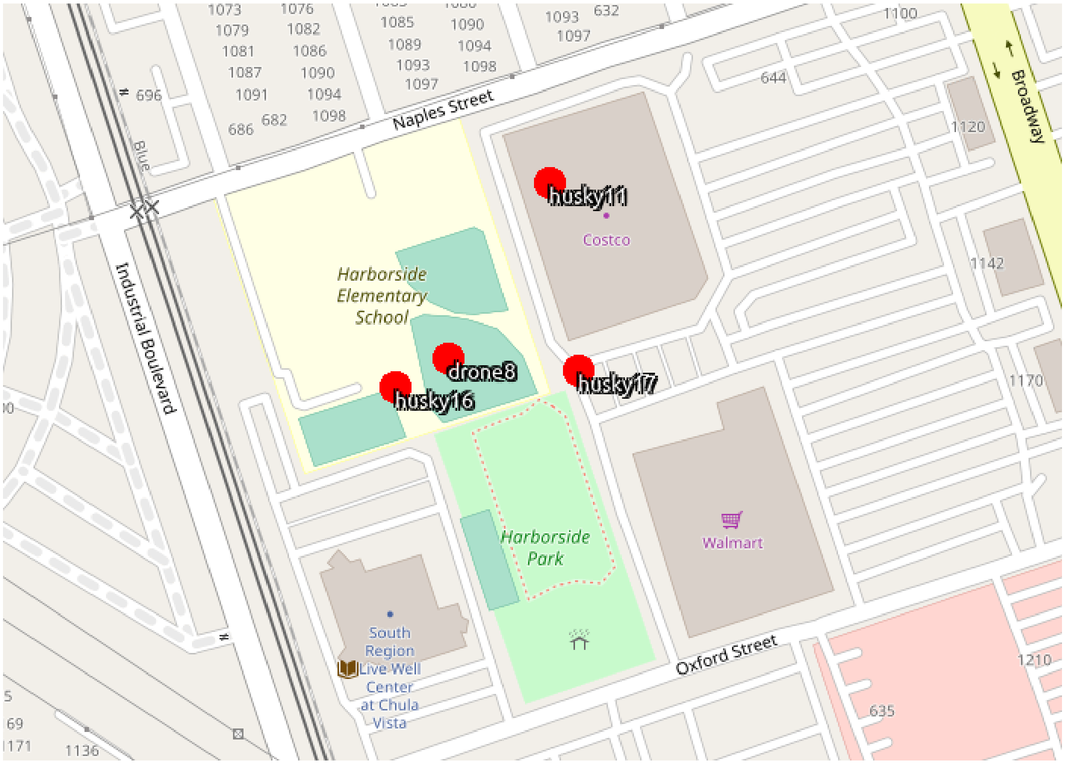


Figure 4.3: Landmark Grounding illustration example.

the gold-standard coordinate. Most subjects used the textual names of the landmarks on the map to refer to them, e.g. instead of other discriminative attributes, such as the colour or type of landmark (e.g. "coffee shop", "supermarket"). They also used robot names themselves as landmarks, instead of just OSM objects. Examples include the following where the landmark is underlined:

Ex 1: *send husky11 62m to the west direction.*

Ex 2: *send 2 drones near Harborside Park (See Figure 4.3).*

USER: Send drone to southern end of pond22.

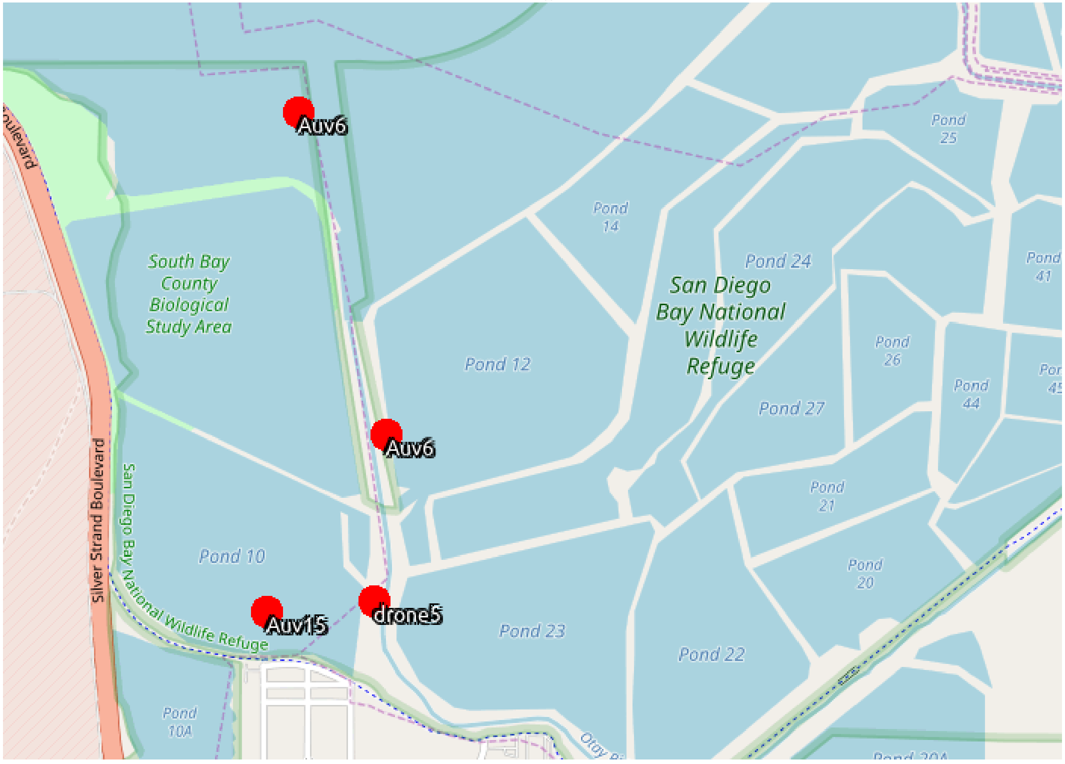


Figure 4.4: Spatial Reference illustration example.

2. **Spatial References:** There are also cases where objects are more than just single point reference GPS coordinates (e.g. polygons, lines) or that subjects use more than one object to clarify the gold-standard end state of the task. For example:

   Ex 3: *send drone to southern end of pond22 (See Figure 4.4).*

   Ex 4: *Move husky land robots to extinguish the fire in between the 685 area and navy Federal credit union.*

USER: Auv17 move <u>1200 meters</u> north and then <u>300 meters</u> east and inspect area.

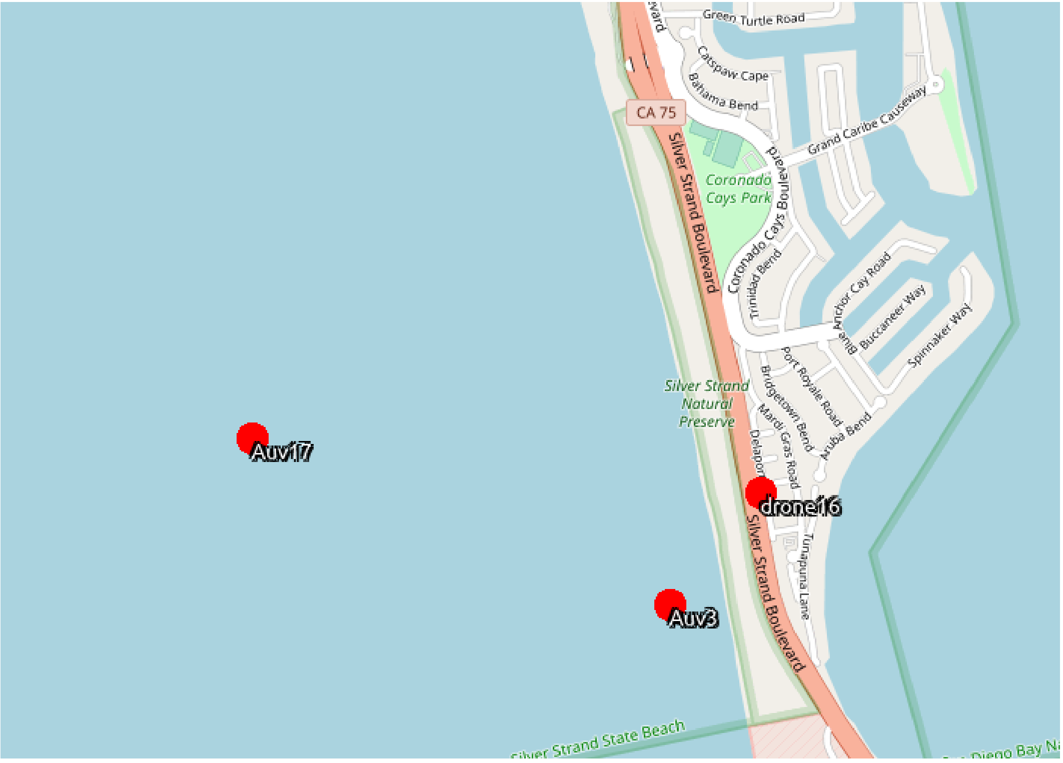


Figure 4.5: Entity Recognition illustration example.

3. **Distance/Bearing:** Entities in the sentence such as numbers (e.g. 500 meters) and directions (e.g. south, NE) were identified. In some cases, prepositions are used instead of numbers (e.g. near, between) or multiple distances which, in turn, need to be grounded to the right landmarks.

   Ex 5: *Auv17 move <u>1200 meters</u> north and then <u>300 meters</u> east and inspect area (See Figure 4.5).*

   Ex 6: *Send a drone <u>near</u> the Silver Strand Preserve.*

Below we show an example of a datapoint, for the instruction "*Send a drone 68m south west of chevron*" together with the collected annotations (landmark, bearing, distance etc.):

```
{
"sentids": [58],                                 %% ID of datapoint
"sentences": [
{
    "raw": "Send a drone 68m south west of chevron",
    "imgid": "3K9FOBBF2KRGR67YZJUHOPAT7D8LNM_1",
    "tokens": ["Send", "a", "drone", "68", "m", "south", \\
    "west", "of", "chevron"], "sentid": 58, "imgid": "3K9FOBBF2"}],
    "image_filename": "3K9F7D8LNM_1.png",
    "scenario_items": "scenario0.json",         %% map/scenario metadata
    "landmarks": [{
        "name": "Chevron",                      %% handlabelled
%        "distance from emergency zone": "68",  %% handlabelled
        "bearing from emergency zone": "225",   %% handlabelled
        "confidence": "2",                      %% handlabelled
        "raw GPS of landmark": [-234879,2898],  %% from OSM
        "geometry type": "Point",               %% from OSM
        "keywords": "convenience shop ",        %% from metadata
        "GPS": [...],
        "stage GPS": [...]                      %% hand annotated
    }]
    ,
    "dynamo_obj": [
                     "drone0_3K8LNM_1",....
                    ],
     human annotated GPS: []                    %% handlabelled
     gold GPS location : []                     %% automatically derived
}
```

Measures	**ROSMI**
# Instructions/Images	783
# Tokens	7,359
# Unique tokens	359
Tokens/instr	9.39 (4-23)
Landmark/instr	1.04 (1-4)
Lexical Sophistication (LS)	0.51
Lexical Density (LD)	0.79
TTR/MSTTR	0.05/0.49

Table 4.1: Lexical Sophistication (LS), Lexical Density (LD), Type-Token Ratio (TTR) and Mean Segmental Type-Token Ratio (MSTTR). Values in parenthesis represent the minimum and maximum value.

4.1.5 Corpus Analysis

Table 4.1 summarises the main descriptive statistics of the ROSMI dataset. We used spaCy[6] tokenizer to split the sentences into tokens resulting in 7,359 tokens. Analysis of the text, showed that 16% of the tokens are numbers or robot names containing numbers (e.g. 500m, husky1, robot34, husky2). In order to get a more accurate number of the unique tokens that cover the vocabulary, we replaced all numbers with the label “NUM” and removed the digits from the robot names (e.g. husky23 = husky). This gave us a total of 359 unique tokens. Each instruction has an average of 9.39 tokens, with a minimum of 4 and a maximum of 23 [9.39(4-23)]. This reflects diversity of instructions and different preferences between subjects on how to approach the task. Furthermore, the average landmark count per instruction (Landmark/instr) is 1.04 with a minimum of 1 and a maximum of 4. Given that the standard deviation is only 0.26, most subjects preferred to use only one landmark to perform the task. We explored the distribution of bigrams and trigrams in our dataset and we show the top 20 in Figure 4.6. 69% of trigrams and 59% of bigrams are used only once in the dataset, further exhibiting diversity in language. The bigram “NUM m” acts as an outlier with a frequency of 527; this is to be expected, as in the majority of the instructions, the bigram “NUM m” is used to represent the distance to the gold-standard coordinate, e.g.˜“500m”.

[6] https://spacy.io/

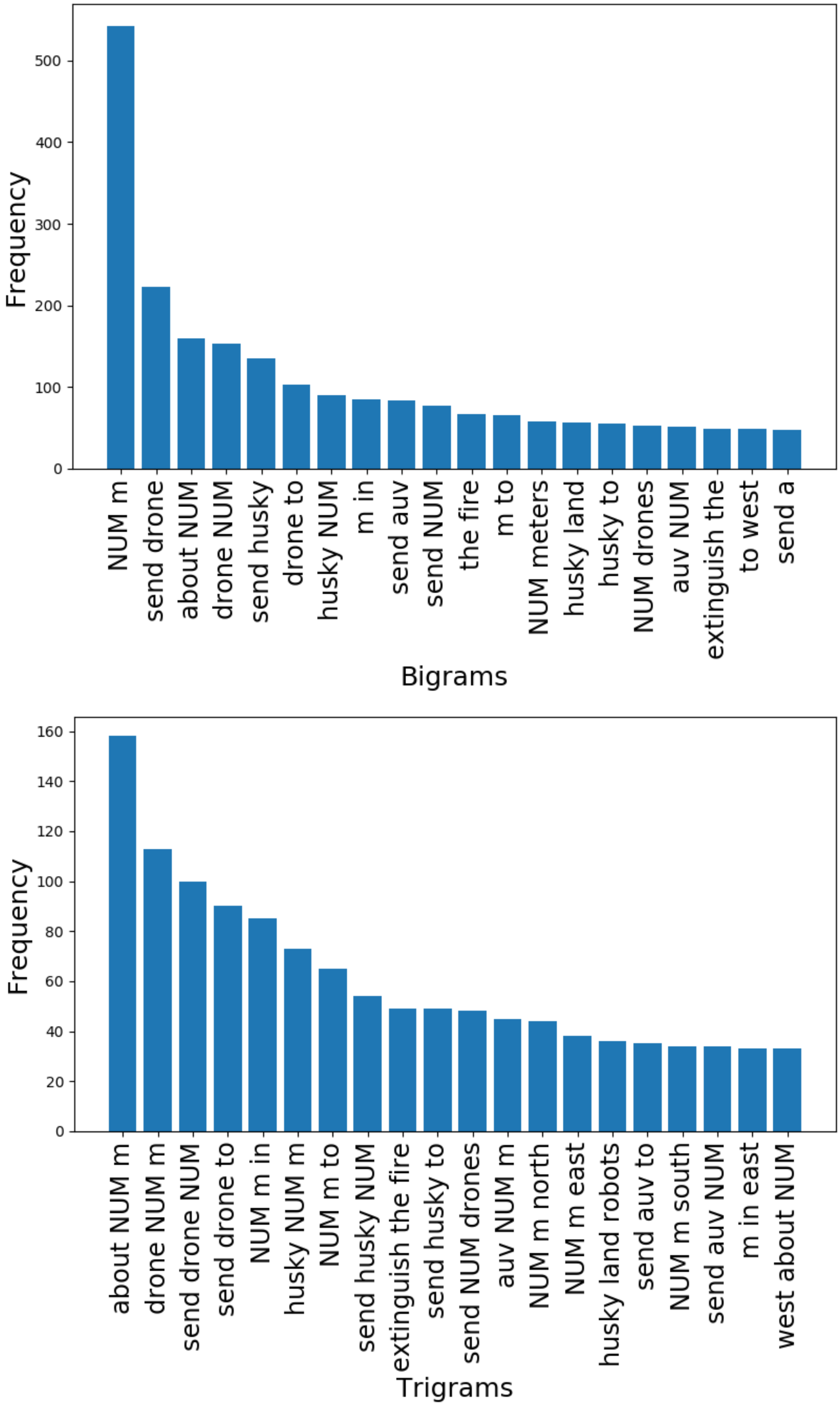


Figure 4.6: The top 20 most frequent, from a long tail of one-off bigrams and trigrams in our dataset (*Left*: distribution of bigrams, *Right*: distribution of trigrams).

4.1.6 Lexical Richness

Lexical richness is a collection of measures on the direct relationship between the number of terms and words used in a text and the diversity of the vocabulary. It has been extensively used to evaluate the lexical diversity in datasets, such as E2E (Novikova et al., 2017) and BAGEL (Mairesse et al., 2010).

Lexical sophistication (LS), also known as lexical rareness, measures the percentage of relatively rare or advanced word types in the text. To determine LS, we measure it against a large corpus of the language of interest. In our case, it is the percentage of lexical types, which are not in the list of 2,000 most frequent words generated from the British National Corpus (BNC). The LS of our corpus is around 50%, measured using the Lexical Complexity Analyser (Ai and Lu, 2010) which is reasonably high and compares well with the 57% LS of the E2E (Novikova et al., 2017), a substantially larger dataset of 50k instances.

Lexical diversity (Torruella and Capsada, 2013) is another aspect of "lexical richness" and refers to the ratio of different unique types that make up our vocabulary, to the total number of words in the corpus. It can be quantitatively measured using several metrics such as the Type-Token Ratio (TTR) and the more robust Mean Segmental Type-Token Ratio (MSTTR), which does not get affected by the text's length, as it splits it into segments of equal length. A higher value of MSTTR means a more diverse text. Our 49% MSTTR shows a fair amount of lexical diversity compared to BAGEL's 42% (Mairesse et al., 2010), but not as much as E2E's 75% (Novikova et al., 2017). E2E is a larger dataset with 65,710 tokens and a larger number of different task specifications (compared to our seven unique tasks), which were defined by more than 50k combinations of a dialogue-act-based meaning representations, giving room for more diverse texts.

Lexical density (LD) (Johansson, 2008) is a measure of how much information a text has. It is calculated by describing the proportion of content words such as nouns, adjectives or verbs to the total number of words, so the more content words there are compared to function words, such as prepositions and pronouns, the more information can be passed on. An LD of 79%, measured against the nouns of our corpus, infers that the instructions carry dense information that could not be summarised further.

4.2 Robot Electronic Navigational Chart Instructions (RENCI)

In Chapter 3, we were interested in giving instructions to AUVs and so the visual stimulus was a C2 interface called SeeTrack (Lane et al., 2013b; Miguelanez et al., 2011; Petillot et al., 2009), which can overlay on top of ENCs. These charts are similar to the OSM, as they are highly complex with rich metadata of entities, but are not common and usually only marine officers can interpret them.

Our goal is to synthesise a dataset called RENCI, by generating textual commands for emergency response, similar to ROSMI, with the same annotations that are described in Section 4.1.4, but with ENCs as maps and not OSMs. We want to be able to generate real-world coordinates given a natural language reference or description of an item, so as to create a robust system and evaluate it with real human experts in Chapter 7.

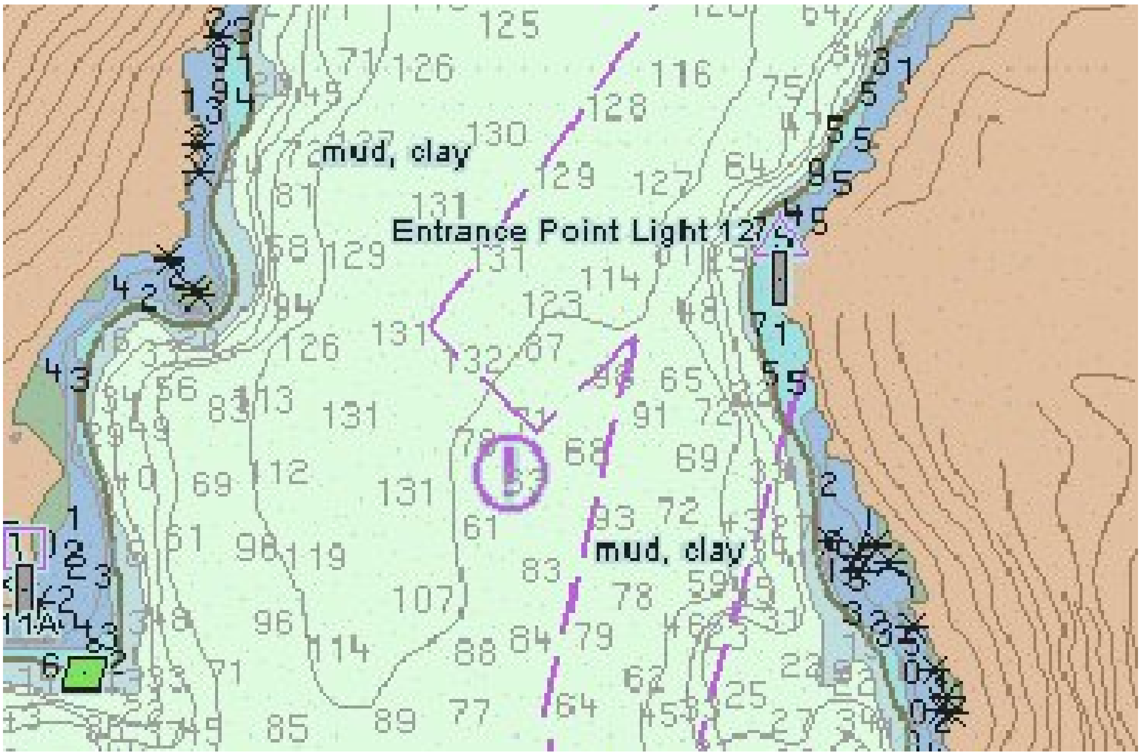


Figure 4.7: Example ENC Chart (Source: `https://en.wikipedia.org/wiki/Electronic_navigational_chart`).

4.2.1 ENC Charts

An ENC (see Figure 4.7) is an official database created by a national hydrographic office for use with an Electronic Chart Display and Information System (ECDIS). An electronic chart must conform to standards stated in the International Hydrographic Organization (IHO) Publication **S-57**[7] before it can be certified as an ENC.

ENCs are available for wholesale distribution to chart agents and resellers from Regional Electronic Navigational Chart Centres (RENCs). Chart data is captured based on standards stated in IHO Publication **S-57**, and is displayed according to a display standard set out in IHO Publication **S-52** to ensure consistency of data rendering between different systems.

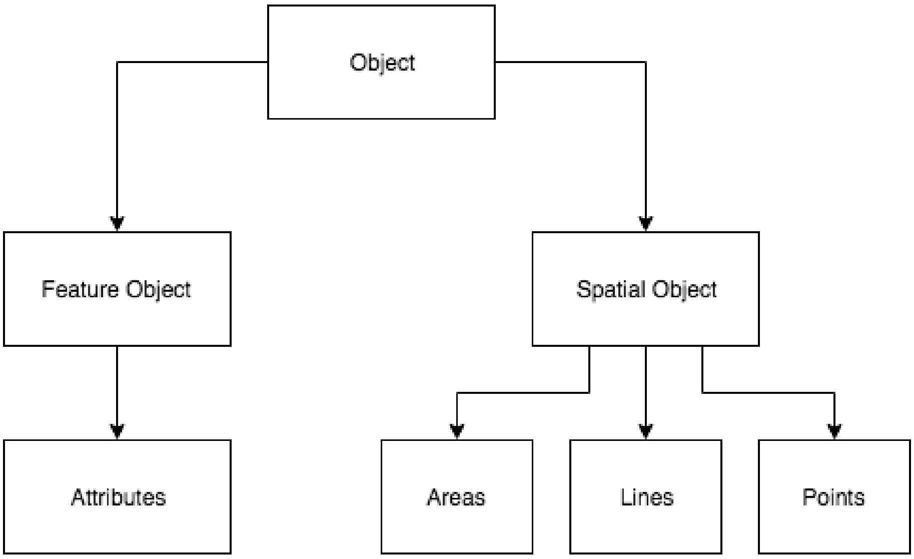


Figure 4.8: S-57 architecture.

4.2.2 S-57 Data Model

The model used in our ENC Charts belongs to the S-57 architecture. In Figure 4.8, we can see a simplified architecture of how the S-57 dataset is structured. There are around 170 **Feature Objects** defined in the S-57 data model, which contain descriptive information. Each of these objects is defined by three sets of attributes, (e.g. colour, information, object name) from a list of 190 sets of attributes. Its object contains **Spatial Objects** as well, which contain positional information, such as latitude, longitude, depth etc. An example is shown in Appendix A.

[7] `http://www.s-57.com/`

4.2.3 Data Generation Method

Given the enormous amount of ENC data, we generated templates using the dialogues (1481 utterances) from our Wizard of Oz experiment (see Chapter 3). For instance, given the sentence "Move the survey 400m south of the shoreline", we generated phrases such as:

Move a surface vehicle 100m north of the 23 flashing light.

Move recovery point 400m east of the chrome bay (see Figure 4.9).

Move survey1 400m west of the fishing facility.

Using the dialogue act "inform" from our WOz data set, we isolated around 600 instructions that provide information or move objects on the map. We further calculated the similarity of each instruction using spaCy's semantic similarity function[8], so as to further confine the types of instruction. From the remaining instructions we carefully selected 32 unique templated sentences (see Appendix A) from the WOz data and by swapping in landmarks, bearings, distances and robots/objects, we generated around 9,000 data points based on seven unique ENCs. These unique charts were chosen with a view to containing a rich set of referable landmarks, with their zoom level adjusted to maximise the visibility of the landmarks. The (ENC

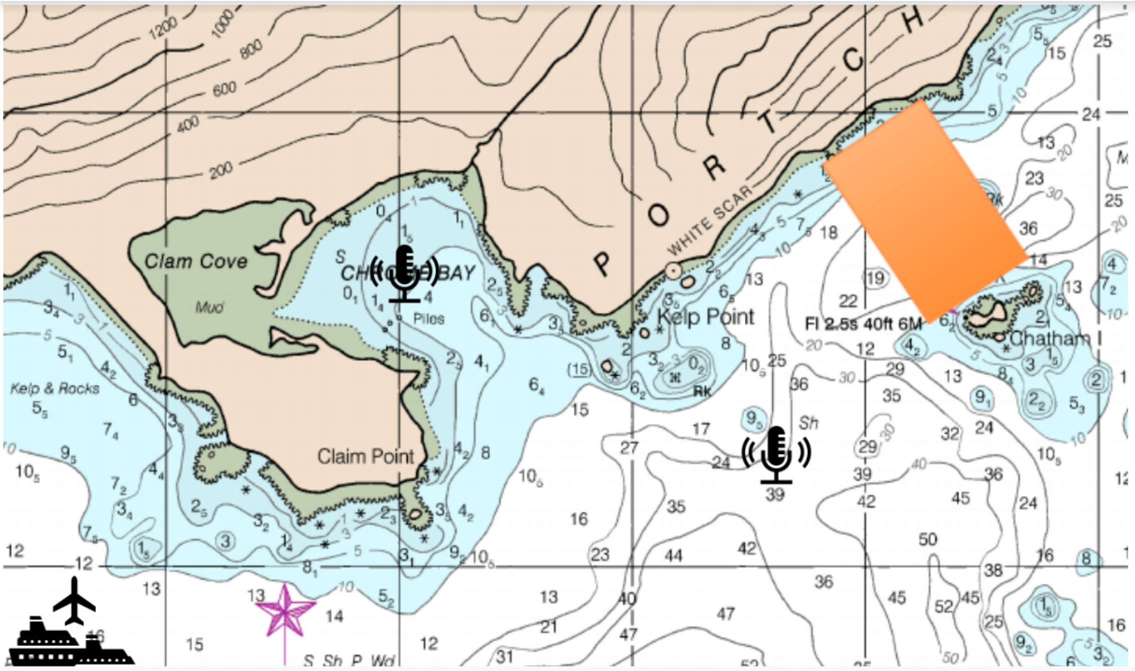


Figure 4.9: One of the seven ENC scenarios for the generation of RENCI.

[8] https://spacy.io/usage/linguistic-features#vectors-similarity

Measures	**ROSMI**	**RENCI**
# Instructions	783	9,139
# Tokens	7,359	106,627
# Unique tokens	359	832
Tokens/instruction	9.39 (4-23)	11.66 (2-26)
Landmark/instruction	1.04 (1-4)	1

Table 4.2: Comparison of the crowdsourced ROSMI dataset and the synthetic dataset RENCI. Values in parenthesis represent the minimum and maximum value.

Context, Sentence) pairs were both in the form of text, compared to ROSMI, which includes images for training as well. These charts have mostly ocean coverage and the tasks are also distinct from those of ROSMI.

We use three types of robots and two types of areas: a) Autonomous Underwater Vehicles (AUVs), b) Unmanned Surface Vehicles (USVs/Boats) and c) Unmanned Aerial Vehicles (UAVs/drones) to rescue vessels in distress; and for searching an area, d) "survey area" and for excluding an area, e) "exclusion zone". Depending on the type of the distress signal, subjects need to localise the vessel in distress on the chart or search an area to locate the vessel in distress or exclude some dangerous area. Each scenario has a unique target GPS location, as in ROSMI, and is used to predict GPS in Chapter 6.

Table 4.2 compares ROSMI and RENCI datasets. Despite RENCI being eleven times larger, with 9,139 instructions, the unique tokens (words) are only double in RENCI, reflecting the diversity of a crowdsourced dataset compared to a generated one. The rest of the statistics, tokens per instructions and landmarks are similar..

4.3 Discussion

In this chapter, we introduced ROSMI, a rich multi-modal dataset comprising 783 pairs of natural language instructions-maps with accompanying metadata, and additional manual annotations, for training end-to-end systems. It is meant to be used for human-robot interaction tasks, with geographical maps as context. The main goal is to be able to predict automatically GPS coordinates as target locations for RAS. This dataset is freely available and we believe that it is a useful addition to studying the problem of grounding language in the real world through efficient and cost-effective crowdsourced data collections. Understanding maps and interacting

Name	**Features**	**Turn-based**	**Number Utterances**	**Usage**	**Described in**
WOz	Text + Vision	Yes	1481	Generating synthetic data	Chapter 3
ROSMI	Text + Vision	No	783	Training/testing MAPERT and hybrid-MAPERT	Chapter 4
RENCI	Text	No	9139	Training/testing MAPERT and hybrid-MAPERT	Chapter 4

Table 4.3: Summary of datasets created and used in this thesis.

with them in a natural manner is a demanding task, incorporating multiple dimensions of complexity that need to be decomposed into more manageable tasks. There are several directions we can use ROSMI to tackle the problem of GPS prediction, some of which are landmark grounding, entity recognition or direct prediction of lat/lon locations.

In addition, RENCI was created, a synthetically generated dataset, based on ENCs, and ten times the size of ROSMI. This dataset, despite being synthetic, can bootstrap the training of neural models on low resource domains such as this one (see Table 4.3 for a complete list of the datasets described in this thesis). Following our data generation, in Chapter 6, we train a neural model end-to-end on these data. Then we want to train neural models to first understand and ground geographical relations with natural language in order to incorporate it in a larger interaction system, as described in 7.

In the next chapter, we focus on ROSMI, due to the extra vision modality, and we implement a multi-modal neural attention model for map understanding and GPS prediction.

Chapter 5

Representing and Understanding Maps

The symbol grounding problem (Harnad, 1990) has been largely studied in the context of mapping language to objects in situated simple (MacMahon et al., 2006; Johnson et al., 2017) or 3D photorealistic environments (Kolve et al., 2017; Savva et al., 2019), static images (Ilinykh et al., 2019; Kazemzadeh et al., 2014b), and to a lesser extent on synthetic (Thompson et al., 1993) and real geographic maps (Paz-Argaman and Tsarfaty, 2019; Haas and Riezler, 2016; Götze and Boye, 2016). These tasks usually relate to navigation (Misra et al., 2018; Thomason et al., 2019a) or action execution (Bisk et al., 2018a; Shridhar et al., 2019) and assume giving instructions to an embodied egocentric agent with a shared first-person view. Since most rely on the visual modality to ground natural language (NL), referring to items in the immediate surroundings: They are often less geared towards the accuracy of the final goal destination.

The task we address here is the prediction of the GPS of a goal destination by reference to a map or chart, which is of critical importance in applications, such as emergency response, where specialised personnel or robots need to operate on an exact location (see Figure 5.1 for an example). Specifically, the goal we are trying to predict is in terms of: a) the GPS coordinates (latitude/longitude) of a referenced landmark; b) a compass direction (bearing) from this referenced landmark; and c) the distance in metres from the referenced landmark. This is done by taking as input into a model: i) the knowledge base of the symbolic representation of the world such

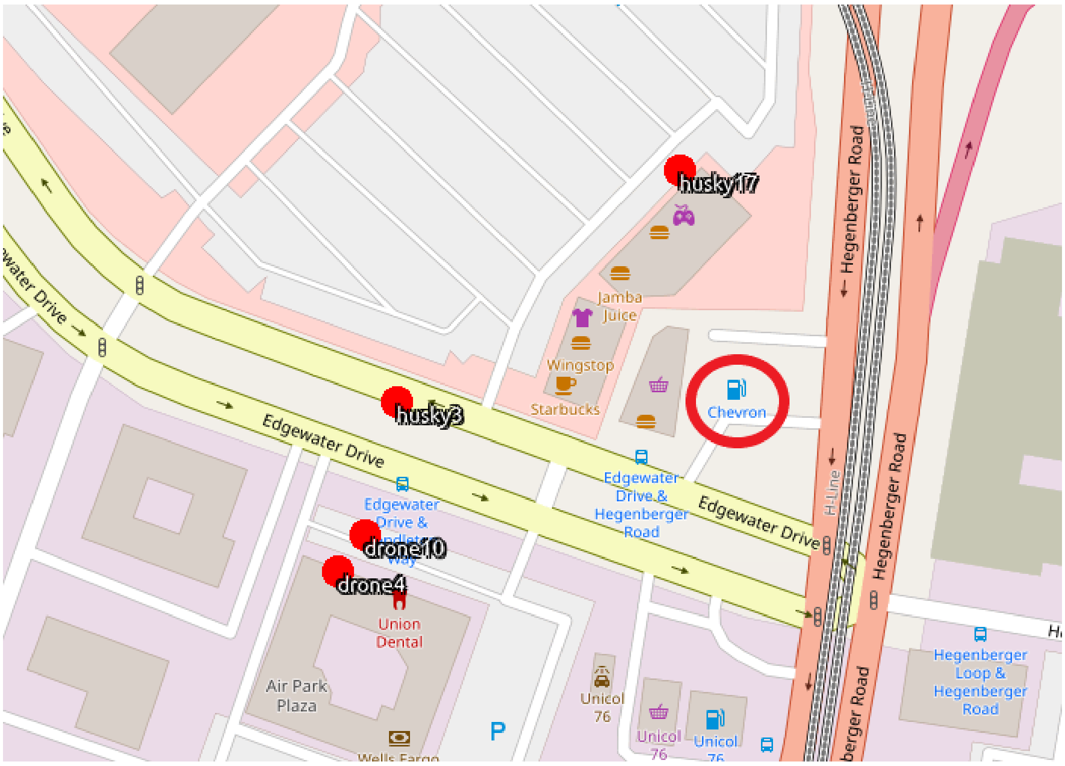


Figure 5.1: Subject instruction and the corresponding image, displaying 4 robots and landmarks. The GPS of the red circled target is what the model is trying to predict.

as landmark names and regions of interest (metadata); ii) the graphic depiction of a map (visual modality); and iii) a worded instruction.

Our approach to the destination prediction task is two-fold. The first stage is the data collection, in Chapter 4, for the “Robot Open Street Map Instructions” (ROSMI) corpus based on OpenStreetMap (Haklay and Weber, 2008), in which we gather and align NL instructions to their corresponding target destination. Whilst OSM and other crowdsourced resources are hugely valuable, there is an element of noise associated with the metadata collected in terms of the names of the objects on the map, which can vary for the same type of object (e.g. newsagent/kiosk, confectionary/chocolate store etc.). Whereas the symbols on the map are from a standard set, which one hypothesises a vision-based trained model could pick-up on. To this end, we developed a model that leverages both vision and metadata to process the NL instructions. The second stage is our Map Encoder Representations from Transformers (MAPERT), a Transformer-based model based on LXMERT, which we train and evaluate on ROSMI. It comprises of up to three single-modality encoders for each input (i.e. vision, metadata and language). An early fusion of these modality components and a cross-modality encoder, which fuses the map representation (metadata and/or vision) with the word embeddings of the instruction in both directions, in order to predict the three outputs, i.e. reference landmark location on the map, bearing and distance.

Experiments on the ROSMI corpus explore a) the importance of modelling both visual and metadata modalities in order to accurately predict target destinations especially for dense maps containing multiple visually identical landmarks, and b) the importance of early fusion of metadata with visual features on a more challenging zero-shot partition of our dataset. We perform an extensive ablative study and error analysis highlighting strengths and weaknesses on variants of our models. The code and data are available at `https://github.com/marioskatsak/mapert`.

The work in this chapter is described and analysed as it was exhibited in the Second International Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP) at the Association for Computational Linguistics (ACL) 2021 (Katsakioris et al., 2021).

The main contribution presented in this chapter is the MAPERT model that is able to understand instructions referring to previously unseen maps and predict GPS goal locations from a map-based natural language instruction (RQ2).

The chapter is organised as follows.Section 5.1, formulates the task and proposes a large attention model, i.e. MAPERT to solve it. Section 5.2, presents the experimental methodology, such as, implementation details and evaluation metrics. Section 5.3, takes the highest performing ablations and shows the overall results of MAPERT and Section 5.4, sums up the chapter.

5.1 Approach

In this section, the task is described in detail, together with MAPERT's architecture and training.

5.1.1 Task Formulation

An instruction is taken as a sequence of word tokens $\mathbf{w} =< w_1, w_2, \dots w_N >$ with $\mathbf{w_i} \in V$, where V is a vocabulary of words, and the corresponding geographic map

I is represented as a set of M landmark objects $o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{n})$ where $\mathbf{bb}$ is a 4-dimensional vector with bounding box coordinates, $\mathbf{r}$ is the corresponding 2048-dimensional Region of Interest (RoI) feature vector, as in (Tan and Bansal, 2019)[1], produced by an object detector and $n =< n_1, n_2 \dots n_K >$, is a multi-token name. We define a function $f : V^N \times \mathbb{R}^{4*M} \times \mathbb{R}^{2048*M} \times V^{M*K} \rightarrow \mathbb{R} \times \mathbb{R}$ to predict the GPS destination location $\hat{y}$:

$$\hat{y} = f\big(\mathbf{w}, \{o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{n})\}_M\big) \tag{5.1}$$

Since predicting $\hat{y}$ directly from $\mathbf{w}$ is a harder task, we decompose it into three simpler components, namely predicting a reference *landmark* location $l \in M$, the compass direction (bearing) b[2], and a distance d from l in metres. Then we trivially convert to the final GPS position coordinates. Equation 5.1 now becomes:

$$\hat{y} = gps(l, d, b) = f\big(\mathbf{w}, \{o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{n})\}_M\big) \tag{5.2}$$

5.1.2 Model Architecture

Inspired by LXMERT (Tan and Bansal, 2019), we present MAPERT, a Transformer-based (Vaswani et al., 2017a) model with three separate single-modality encoders (for NL instructions, metadata and visual features) and a cross-modality encoder that merges them. Figure 5.2 depicts the architecture. In the following sections, we describe each component separately.

Instructions Encoder The word sequence $\mathbf{w}$ is fed to a Transformer encoder and output hidden states $\mathbf{h_w}$ and position embeddings $\mathbf{pos_w}$; its weights are initialised using pretrained BERT (Devlin et al., 2019). $\mathbf{h_{w_0}}$ is the hidden state for the special starting token [CLS].

Metadata Encoder OSM comes with useful metadata in the form of bounding boxes and names of landmarks on the map. We represent each bounding box as a 4-dimensional vector $\mathbf{bb_{meta_k}}$ and each name ($\mathbf{n_k}$) using another Transformer

[1] A 1024-dimensional RoI feature vector was also tested but without producing any better results, therefore keeping the 2048 dimensions.

[2] $b \in \{N, NE, NW, E, SE, S, SW, SE, W, None\}$.

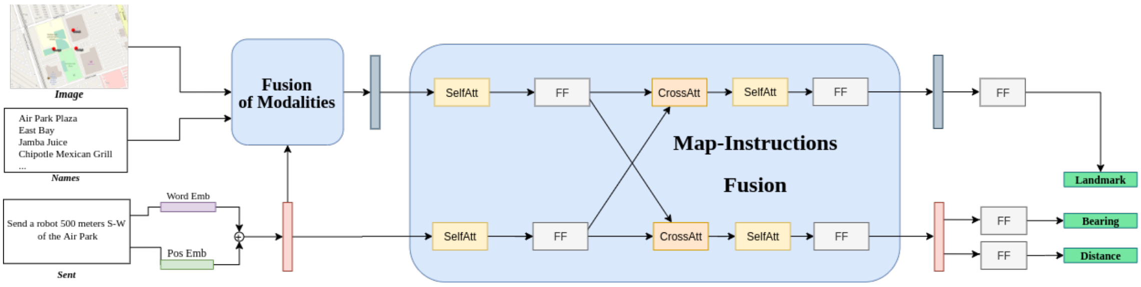


Figure 5.2: Architecture of MAPERT. Map representations, i.e. names of landmarks found in OSM (metadata) and Faster-RCNN predicted objects (visual modality), along with an instruction (sequence of tokens) are a) encoded into the model, b) fused together (see also Figure 5.4) and c) bidirectionally attended. The output comprises of three predictions, recast as classification tasks: a landmark, a bearing and a distance.

initialised with pretrained BERT weights. We treat metadata as a bag of names but since each word can have multiple tokens, we output position embeddings $\mathbf{pos_{n_k}}$ for each name separately; $\mathbf{h_{n_k}}$ are the resulting hidden states with $\mathbf{h_{n_{k,0}}}$ being the hidden state for [CLS].

Visual Encoder Each map image is fed into a pretrained Faster R-CNN detector (Ren et al., 2015b), which outputs bounding boxes and RoI feature vectors $\mathbf{bb_k}$ and $\mathbf{r_k}$ for k objects. In order to learn better representation for landmarks, we fine-tuned the detector on around 27k images of maps to recognise k objects $\{o_1, .., o_k\}$ and classify landmarks of 213 manually-cleaned classes from OSM; we fixed k to 73 landmarks. Finally, a combined position-aware embedding was learned $\mathbf{v_k}$ by adding together the vectors $\mathbf{bb_k}$ and $\mathbf{r_k}$, as in LXMERT:

$$\mathbf{v_k} = \frac{FF(\mathbf{bb_k}) + FF(\mathbf{r_k})}{2} \tag{5.3}$$

where FF are feed-forward layers with no bias.

5.1.3 Variants for Fusion of Input Modalities

We describe three different approaches to combining knowledge from maps with the NL instructions:

Metadata and Language The outputs of the metadata and language encoders are fused by conditioning each landmark name n_i on the instruction sequence via

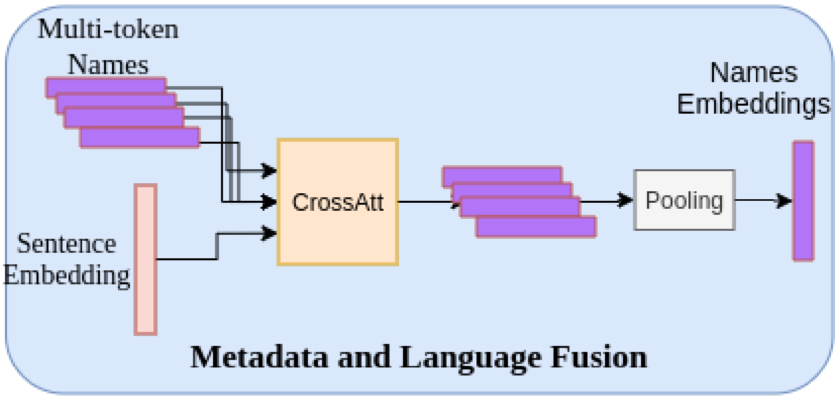


Figure 5.3: Metadata and Language fusion module. Multi-token names correspond to the BERT-based embeddings of landmarks names. The output is the embedding used to represent the landmarks names from OSM metadata.

a uni-directional cross-attention layer (Figure 5.3). We first compute the attention weights A_k between the name tokens $\mathbf{n_{k,i}}$ of each landmark o_k and instruction words in $\mathbf{h_w}$[3] and re-weight the hidden states $\mathbf{h_{n_k}}$ to get the context vectors $\mathbf{c_{n_k}}$. We then pool them using the context vector for the [CLS] token of each name:

$$\mathbf{A_k} = CrossAttn(\mathbf{h_w}, \mathbf{n_k}) \tag{5.4}$$

$$\mathbf{c_{n_k}} = \mathbf{A_k} \odot \mathbf{n_k} \tag{5.5}$$

$$\mathbf{h_{meta}} = BertPooler(\mathbf{c_{n_k}}) \tag{5.6}$$

We can also concatenate the bounding box $\mathbf{bb_{meta_k}}$ to the final hidden states:

$$\mathbf{h_{meta+bb}} = [\mathbf{h_{meta}}; FF(\mathbf{bb_{meta_k}})] \tag{5.7}$$

Metadata+Vision and Language All three modalities were fused to verify whether vision can aid metadata information for the final GPS destination prediction task (Figure 5.4). First, we filter the landmarks o_i based on the Intersection over Union between the bounding boxes found in metadata ($\mathbf{bb_{meta_k}}$) and those predicted with Faster R-CNN ($\mathbf{bb_k}$), thus keeping their corresponding names n_i and visual features $\mathbf{v_i}$. Then, we compute the instruction-conditioned metadata hidden states $\mathbf{h_{meta_i}}$, as described above, and multiply them with every object v_i to get the

[3] Whenever we refer to hidden states $\mathbf{h_w}$, we assume concatenation with corresponding positional embeddings $[\mathbf{h_w}; \mathbf{pos_w}]$.

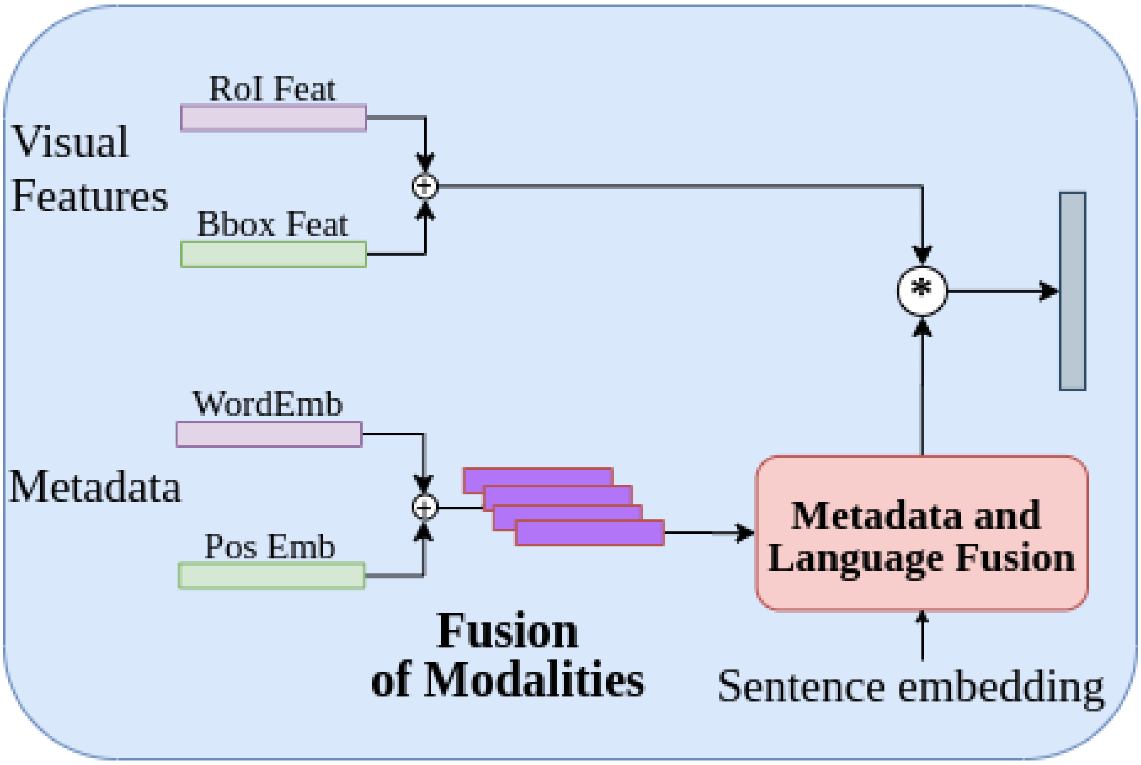


Figure 5.4: Fusion of metadata, vision and language modalities. Metadata are first conditioned on the instruction tokens as shown in Figure 5.3. They are subsequently multiplied with the visual features of every landmark.

final $\mathbf{h_{meta+vis}}$ context vectors:

$$\mathbf{h_{meta+vis_i}} = \mathbf{h_{meta_i}} \otimes \mathbf{v_i} \tag{5.8}$$

5.1.4 Map-Instructions Fusion

So far we have conditioned modalities in one direction, i.e. from the instruction to metadata and visual features. In order to capture the influence between map and instructions in both ways, a cross-modality encoder was implemented (right half of Figure 5.2). Firstly each modality passes through a self-attention and feed-forward layer to highlight inter-dependencies. Then these modulated inputs are passed to the actual fusion component, which consists of one bi-directional cross-attention layer, two self-attention layers, and two feed-forward layers. The cross-attention layer is a combination of two unidirectional cross-attention layers, one from instruction tokens ($\mathbf{h_w}$) to map representations (either of $\mathbf{h_{meta_k}}$, $\mathbf{v_k}$ or $\mathbf{h_{meta+vis_k}}$; we refer to them below as $\mathbf{h_{map_k}}$) and vice-versa:

$$\tilde{\mathbf{h}}_{\mathbf{w}} = FF(\mathit{SelfAtt}(\mathbf{h}_{\mathbf{w}})) \tag{5.9}$$

$$\tilde{\mathbf{h}}_{\mathbf{map_k}} = FF(\mathit{SelfAtt}(\mathbf{h}_{\mathbf{map_k}})) \tag{5.10}$$

$$\mathbf{C}_{\mathbf{map_k}} = \mathit{CrossAtt}(\tilde{\mathbf{h}}_{\mathbf{w}}, \tilde{\mathbf{h}}_{\mathbf{map_k}}) \tag{5.11}$$

$$\mathbf{C}_{\mathbf{w}} = \mathit{CrossAtt}(\tilde{\mathbf{h}}_{\mathbf{map_k}}, \tilde{\mathbf{h}}_{\mathbf{w}}) \tag{5.12}$$

$$\mathbf{h}_{\mathbf{cross,w}} = \mathbf{C}_{\mathbf{w}} \odot \tilde{\mathbf{h}}_{\mathbf{w}} \tag{5.13}$$

$$\mathbf{h}_{\mathbf{cross,map_k}} = \mathbf{C}_{\mathbf{map_k}} \odot \tilde{\mathbf{h}}_{\mathbf{map_k}} \tag{5.14}$$

$$\mathbf{out}_{\mathbf{w}} = FF(\mathit{SelfAtt}(\mathbf{h}_{\mathbf{cross,w}})) \tag{5.15}$$

$$\mathbf{out}_{\mathbf{map_k}} = FF(\mathit{SelfAtt}(\mathbf{h}_{\mathbf{cross,map_k}})) \tag{5.16}$$

Note that representing $\mathbf{h}_{\mathbf{map_k}}$ with vision features $\mathbf{v_k}$ only, is essentially a fusion between the vision and language modalities. This is a useful variant of our model to measure whether the visual representation of a map alone is as powerful as metadata, specifically for accurately predicting the GPS location of the target destination.

5.1.5 Output Representations and Training

As shown in the right-most part of Figure 5.2, our MAPERT model has three outputs: landmarks, distances, and bearings. We treat each output as a classification sub-task, i.e. predicting one of the k landmarks in the map; identifying in the NL instruction the start and end position of the sequence of tokens that denotes a distance from the reference landmark (e.g. ‘*‘500m”*); and a bearing label. MAPERT's output comprises of two feature vectors, one for the vision and one for the language modality generated by the cross-modality encoder.

More specifically, for the bearing predictor, we pass the hidden state $\mathbf{out}_{\mathbf{w},\mathbf{0}}$, corresponding to [CLS], to a FF followed by a softmax layer. Predicting distance is similar to span prediction for Question Answering tasks; we project each of the tokens in $\mathbf{out}_{\mathbf{w}}$ down to 2 dimensions corresponding to the distance span boundaries in the instruction sentence. If there is no distance in the sentence e.g. *“Send a drone* *at Jamba Juice"*, the model learns to predict, both as start and end position, the final 'end of sentence' symbol, as an indication of absence of distance. Finally, for landmark prediction we project each of the k map hidden states $\mathbf{out_{map_k}}$ to a single dimension corresponding to the index of the i^{th} landmark.

We optimise MAPERT by summing the cross-entropy losses for each of the classification sub-tasks. The final training objective becomes:

$$\mathcal{L} = \mathcal{L}_{land} + \mathcal{L}_{bear} + \mathcal{L}_{dist,start} + \mathcal{L}_{dist,end} \tag{5.17}$$

5.2 Experimental Setup

Implementation Details We evaluate our model on the ROSMI dataset and assess the contribution of the metadata and vision components as described above. For the attention modules, we use a hidden layer with size of 768 as in $BERT_{BASE}$ and we set the numbers of all the encoder and fusion layers to 1. We initialise pretrained BERT embedding layers (we do not show results with randomly initialised embeddings due to their clear disadvantage). We trained our model using Adam (Kingma and Ba, 2015) as the optimizer with a linear-decayed learning-rate schedule (Tan and Bansal, 2019) for 90 epochs, a dropout probability of 0.1 and learning rate of 10^{-3}.

Evaluation Metrics We use a 10-fold cross-validation for our evaluation methodology. This results in a less biased estimate of the model skill over splitting the data into train/test, due to the modest size of the dataset. In addition, we performed a leave-one-map-out cross-validation, as in Chen and Mooney (2011). In other words, we use 7-fold cross-validation, and in each fold we use six maps for training and one map for validation. We refer to these scenarios as zero-shot[4] since, in each fold, we validate our data on an unseen map scenario. With the three outputs of our model, landmark, distance and bearing, we indirectly predict the destination location. Success is measured by the Intersection over Union (IoU) between the ground truth destination location and the calculated destination location. IoU measures

[4] We broadly use the term zero-shot as we appreciate that there might be some overlap in terms of street names and some objects.

the overlap between two bounding boxes and as in Everingham et al. (2010), must exceed 0.5 (50%) to count it as successful by the formula:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{5.18}$$

Since we are dealing with GPS coordinates but also image pixels, we report two error evaluation metrics. The first is sized weighted Target error (T err) in metres, which is the distance in metres between the predicted GPS coordinate and the ground truth coordinate. The second is a Pixel Error (P error) which is the difference in pixels between the predicted point in the image and the ground truth converted from the GPS coordinate.

Comparison of Systems We evaluate our system on three variants using different fusion techniques, namely Meta and Language; Meta+Vision and Language; and Vision and Language. Ablations for these systems are shown in Table 5.1 and are further analysed in Section 5.3. We also compare MAPERT to a strong baseline. For the baseline, we use a neural model which encodes the input features similar to MAPERT, but instead of the bidirectional cross-attention layers in the pipeline (see Figure 5.2), uses only feedforward layers for predicting the outputs.

Note, the Oracle of the Meta and Language has a 100% (upper bound) on both cross-validation splits of ROSMI, whereas the oracle of any model that utilises visual features, is 80% in the 10-fold and 81.98% in the 7-fold cross-validation (lower bound). In other words, the GPS predictor can only work with the output of the automatically predicted entities outputed from Faster R-CNN, of which 20% are inaccurate. Table 5.1 shows results on both oracles, with the subscript *lower* indicating the lower bound oracle and *upper* indicating the “Upper Bound” oracle. In Table 5.2, all systems are being projected on the lower bound oracle, so as to compare them on the same footing.

5.3 Results

Table 5.2 shows the results of our model for Vision, Meta and Meta+Vision (M+V) on both the 10-fold cross validation and the 7-fold zero-shot cross validation. We see that the Meta variant of MAPERT outperforms all other variants and our baseline.

	10-fold Cross Validation	
	Acc_{50} [SD]	T err(m) [SD]
Oracle$_{low}$	80 [5.01]	23.8 [51.9]
Vision		
bbox	46.18 [5.59]	44.7 [51.7]
RoI+bbox	60.36 [5.3]	36.4 [51.1]
Meta+Vision		
RoI+bbox+names	69.27 [6.68]	26.9 [47.7]
Meta		
bbox	46.18 [5.59]	44.7 [51.7]
names	**71.81 [7.37]**	26.7 [47.7]
bbox+names	70.73 [6.58]	26.3 [48.7]
Oracle$_{upper}$	100 [0.0]	0 [0]
Meta		
bbox	60.36 [5.26]	29.8 [44.9]
names	**87.64 [4.8]**	9.6 [29.9]
bbox+names	87.09 [5.66]	9.5 [27.2]

Table 5.1: Ablation results on ROSMI using a 10-fold cross validation. Accuracy (Acc) with IoU of 0.5 and Target error (T err) in metres. The results in the top half of the table use names conditioned on the lower bound of the Vision modality and so are compared to Oracle$_{low}$. The bottom part of the table use the true metadata names and so are to be compared to Oracle$_{upper}$.

However, looking at the 10-fold results, Meta+Vision's accuracy of 69.27% comes almost on par with Meta's 71.81%. If we have the harder task of no metadata, with only the visuals of the map to work with, we can see that the Vision component works reasonably well, with an accuracy to 60.36%.

This Vision component, despite being on a disadvantage, manages to learn the relationship of visual features with an instruction and vice-versa, compared to our baseline, which has no crossing between the modalities whatsoever, reaching only 33.82%. When we compare these results to the zero-shot paradigm in Table 5.2, we see only a 7.5% reduction using Meta, whereas the Vision-only and M+V components struggle more, with an around 10% reduction. This is understandable since on the 7-fold validation, we tackle unseen maps, which is very challenging for the Vision component of the models.

A two proportion z-test at significance level $\alpha = 0.05$ and population $n = 55$ of the test data is also performed, so as to examine the significance of the different accuracy scores between the models for Table 5.2 (including the Oracle). All models significantly outperform the Baseline, in both the 10-fold and 7-fold cross validation.

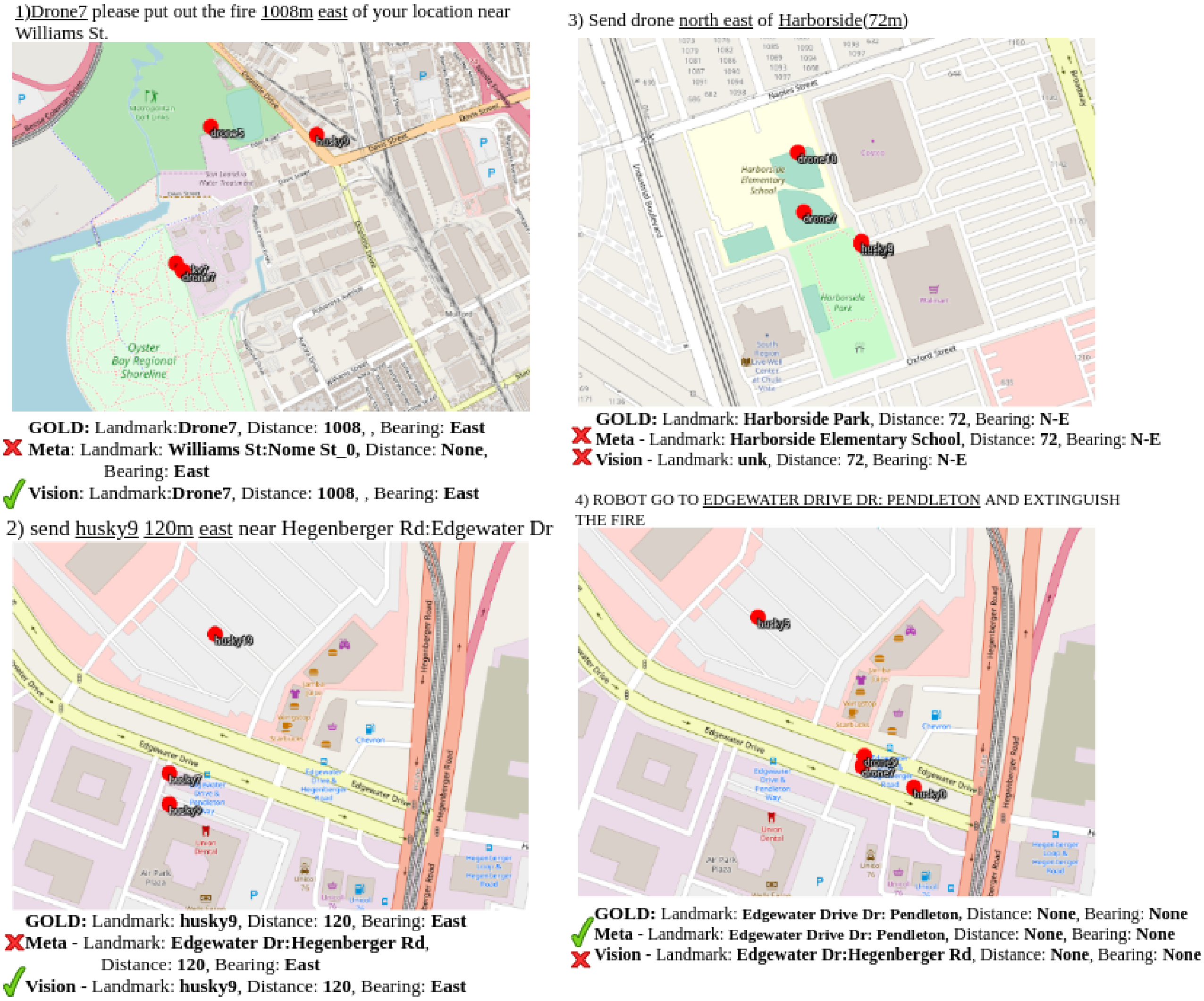
	10-fold Cross Validation (unseen examples)			**7-fold Cross Validation** (unseen scenarios)		
	Acc$_{50}$ [SD]	T err [SD]	P err [SD]	Acc$_{50}$ [SD]	T err (m)[SD]	P err (m) [SD]
Oracle$_{low}$	80 [5.0]	23.8 [51.9]	39.1 [96.3]	82 [17.1]	20.1 [39]	33.3 [66.4]
Baseline	33.8 [5.2]	64 [57.1]	120 [112.3]	34.9 [11.1]	60.7 [57.1]	110.4 [109.7]
Meta	**71.8 [7.4]**	26.7 [47.7]	48.2 [91.2]	**64.3 [14.2]**	32.7 [50.1]	65.7 [88.4]
Vision	60.4 [5.3]	36.4 [51.1]	64.4 [99.6]	49.75 [8.1]	46.0 [54.6]	87.9 [106.0]
M+V	69.3 [6.7]	26.9 [47.7]	48.3 [91.4]	58.3 [12.2]	36.1 [46.1]	70.7 [93.3]

Table 5.2: Results on both cross-validations of the best performing ablations of each variant and the baseline. The predictions have been made under the Oracle$_{low}$. Accuracy (Acc) with IoU of 0.5, Target error (T err) and Pixel error (P err) in metres. The "unseen scenarios" refer to unseen maps, whereas "unseen examples" refers to unseen instructions but on maps seen during training.

Although Meta has the highest accuracy score, there are no significant differences, at level $\alpha = 0.05$, between the mode ablations but one has to be careful of analysing 'non-significance' because one could collect more data and find significance. This indicates that having only one, well-tailored modality on MAPERT, will not make significant difference in the results but more experiments would need to be performed to say this for certain.

Ablation Study We show ablations on the 10-fold as the 7-fold has similar performance ordering, for all three model variants in Table 5.1. Depending on the representation of the map for each variant, we derive three ablations for the Meta and two for the Vision. Meta+Vision does not have ablations, since it stands for all possible representations (bb, r, n). Compared to the Oracle$_{low}$, Meta outperforms the rest, as seen in Table 5.2. In addition, it requires only the names of the landmarks to score the 71.73%. When we fuse the names and the bboxes, the accuracy decreases slightly, whereas the T err decreases slightly from 26.7 metres to 26.3 metres. The full potential of the Meta model is shown on the Oracle$_{upper}$, which reaches 87.64% accuracy and T err of only 9.6 metres, proof that for our task and dataset metadata has the upper hand. It is worthwhile noting that the Vision variant would not have reached 60.36% accuracy, without the r features, since with no fusion of RoI, the accuracy drops to 46.18%.

Error Analysis In order to understand where the Vision and Meta models' comparative strengths lie, we show some example outputs in Figure 5.5. In examples



1)Drone7 please put out the fire 1008m east of your location near Williams St.

GOLD: Landmark:**Drone7**, Distance: **1008**, , Bearing: **East**

✗ **Meta**: Landmark: **Williams St:Nome St_0,** Distance: **None**, Bearing: **East**

✓ **Vision**: Landmark:**Drone7**, Distance: **1008**, , Bearing: **East**

2) send husky9 120m east near Hegenberger Rd:Edgewater Dr

GOLD: Landmark: **husky9**, Distance: **120**, Bearing: **East**

✗**Meta** - Landmark: **Edgewater Dr:Hegenberger Rd**, Distance: **120**, Bearing: **East**

✓ **Vision** - Landmark: **husky9**, Distance: **120**, Bearing: **East**

3) Send drone north east of Harborside(72m)

GOLD: Landmark: **Harborside Park**, Distance: **72**, Bearing: **N-E**

✗ **Meta** - Landmark: **Harborside Elementary School**, Distance: **72**, Bearing: **N-E**

✗ **Vision** - Landmark: **unk**, Distance: **72**, Bearing: **N-E**

4) ROBOT GO TO EDGEWATER DRIVE DR: PENDLETON AND EXTINGUISH THE FIRE

✓**GOLD:** Landmark: **Edgewater Drive Dr: Pendleton,** Distance: **None**, Bearing: **None**

Meta - Landmark: **Edgewater Drive Dr: Pendleton**, Distance: **None**, Bearing: **None**

✗ **Vision** - Landmark: **Edgewater Dr:Hegenberger Rd**, Distance: **None**, Bearing: **None**

Figure 5.5: Examples of instructions with the corresponding maps and the accompanied predictions of the best performing either Vision or Meta models conditioned on Oracle$_{low}$. Underlined words are words corresponding to the target output of the model.

1&2 in this figure, we see the Meta model is failing to identify the correct landmark because the instruction is formulated in a way that allows the identification of two landmarks. It is a matter of which landmark to choose, and the bearing, distance that comes with it, to successfully predict the destination location. However, the Meta model is mixing up the landmarks and the bearings. We believe it is that perhaps the Meta model struggles with spatial relations such as “near”. The Vision model, on the other hand, successfully picks up the three correct components for the prediction. This might be helped by the familiarity of the symbolic representation of the robots (husky (UGV), drones (UAV), AUV), which it is able to pick up and use as landmarks in situations of uncertainty such as this one. Both models can fail in situations of both visual and metadata ambiguity. In the third example, the landmark (Harborside Park) is not properly specified and both models fail to pinpoint the correct landmark, since further clarification would be needed. The final example in Figure 5.5 shows a situation in which the Meta model works well without the need for a specific distance and bearing. The Vision model manages to capture that, but it fails to identify the correct landmark.

5.4 Conclusions

We have developed a model that is able to process instructions on a map using metadata from rich map resources such as OSM and can do so for maps that it has not seen before with only a 10% reduction in accuracy. If no metadata is available then the model can use Vision, although this is clearly a harder task. Vision does seem to help in examples where there is a level of uncertainty such as with spatial relations or ambiguity between entities.

Salin et al. (2022) examine different vision and language (VL) models, such as LXMERT (Tan and Bansal, 2019) and UNITER (Chen et al., 2020), and also, show that Faster-RCNN features can be a limiting factor when performing VL tasks. Despite their ability to extract multi-modal information on some concepts, such as colour, less objective concepts, such as position and size, are harder for these models to grasp. Because of this limitation, these models end up relying more on textual information, similar to our findings. In the next chapter, MAPERT is tested on different tasks and more emphasis is given on the language-oriented part of MAPERT. Different types of input features are examined, especially text metadata and a hybrid version of MAPERT is created that leverages the strength of the language hidden state for predicting landmarks.

Chapter 6

The Robustness of MAPERT on Varying Datasets and New Tasks

This chapter explores the capabilities of MAPERT, which was introduced in Chapter 5, by keeping the core of the model intact and testing it on different datasets, with varied metadata and new tasks (RQ3). Specifically, MAPERT was tested on the Blocks World task[1], a dataset first proposed by Winograd (1972) and more recently released by Bisk et al. (2016a), which explores understanding natural language communication in a situated environment between a human and an agent. We also evaluated MAPERT on RENCI, the synthetic dataset described in Chapter 4, based on ENC charts, which includes expert language and complex structures.

The content presented in this chapter spans across the following three sections:

- Section 6.1 introduces the task of the Blocks World, examines various input metadata and shows the results of an ablation study on MAPERT.
- Section 6.2 tests MAPERT on the synthetic RENCI dataset described in Chapter 4 and adapts the non-parametric BM25 search method to MAPERT, so as to create a hybrid model for predicting GPS goal locations from a map-based natural language.
- Section 6.3 summarises the chapter and sets the ground for the next Chapter, where we explore Human-In-The-Loop learning.

[1] As part of the collaboration with Javier Chiyah Garcia.

6.1 The Blocks World

Similar to ROSMI, the Blocks World (Bisk et al., 2016a) task investigates understanding natural language instructions in a situated environment (an image of blocks rather than a map) between a human and an agent (see Figure 6.1). The human's goal is to arrange blocks a certain way by giving instructions to the agent, which in turns modifies the environment. We focus on the version of the dataset where blocks have no defining features (blank-labelled) and thus need to be referenced with challenging spatial co-references. Unlike other datasets, or the versions of this dataset with numbers and logos in the blocks, blank blocks are much more ambiguous. Thus, the blank blocks version of this dataset is an ideal setting to research understanding natural language instructions in an ambiguous situated environment that cannot be easily solved by an improvement of object recognition or reasoning alone. Refer to Bisk et al. (2016a) for more information about the blocks world dataset.

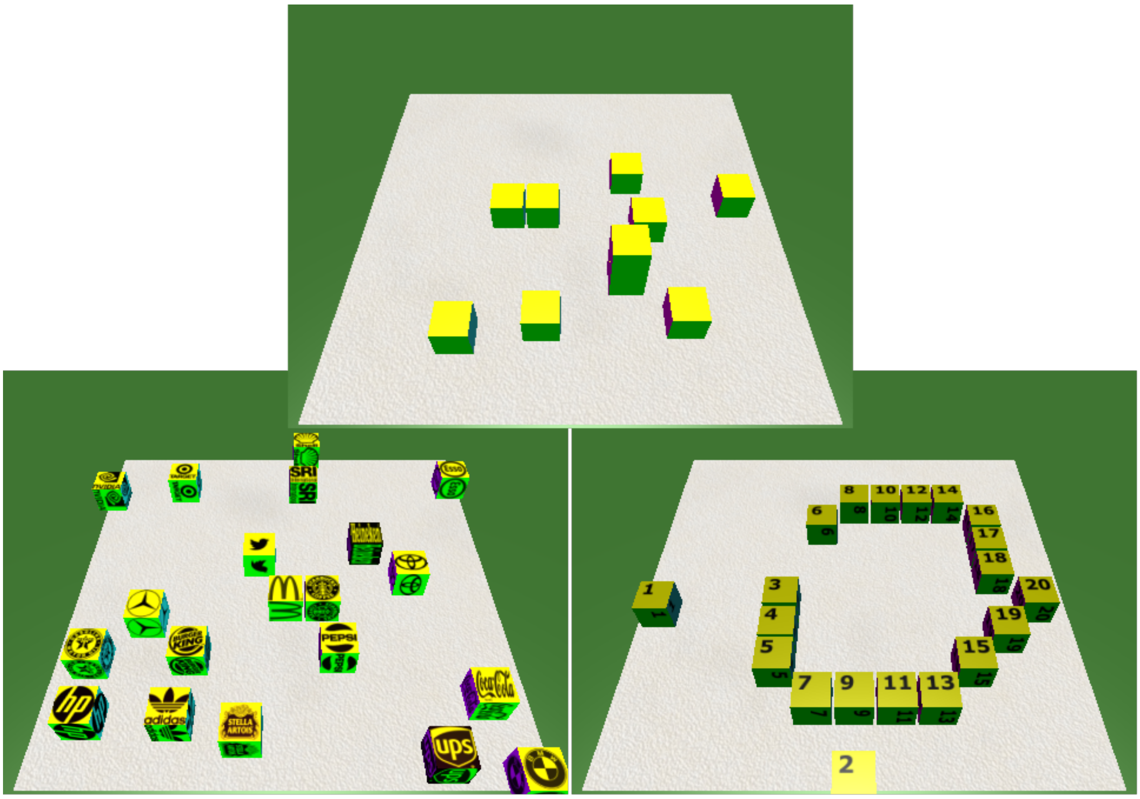


Figure 6.1: Examples of all Blocks World's block types (blank, digit and logos). Source: `https://groundedlanguage.github.io/` .

Solving the scenarios in the dataset requires: (1) predicting the block that needs to be moved (source block); and (2) predicting where to move it (target position). An example instruction from the dataset is *"Move the last block in the last row and put it between the last block in the second row and the stack of blocks."*.

6.1.1 MAPERT for the Blocks World

Similar to Section 5.1, an instruction is taken as a sequence of word tokens $\mathbf{w} =< w_1, w_2, \ldots w_N >$ with $\mathbf{w_i} \in V$, where V is a vocabulary of words. The corresponding image I can be represented as a set of M blocks $o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{wst}, \mathbf{eud}, \mathbf{sen})$ where:

- **bb** is a 4-dimensional vector with bounding box coordinates;
- **r** is the corresponding Region of Interest (RoI) feature vector produced by an object detector;
- **wst** is the world state, a 3-dimensional vector with xyz coordinates where y is the height and is always the same (unless the blocks are stacked);
- **eud** is the euclidean distance to edges and corners, an 8-dimensional vector with four distances (in order, top, right, bottom, left) from the edges of the board and four distances from the corners of the board;
- **sen** is the spatial encoding, a 3-dimensional vector with: (1) one value of whether the block is stacked on top of another one; (2) one value for the East-West split (where a block is on the table, towards the East, West or neither); (3) North-South split (if the block is towards the North, South or neither).

We define a function $f : V^N \times \mathbb{R}^{4*M} \times \mathbb{R}^{2048*M} \times \mathbb{R}^{3*M} \times \mathbb{R}^{8*M} \times \mathbb{R}^{3*M} \rightarrow \mathbb{R} \times \mathbb{R}$ to predict the target position of the block $\hat{y}$:

$$\hat{y} = f\big(\mathbf{w}, \{o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{wst}, \mathbf{eud}, \mathbf{sen})\}_M\big) \tag{6.1}$$

Since predicting $\hat{y}$ directly from $\mathbf{w}$ is a harder task, similarly to ROSMI, we decompose it into three simpler components, namely predicting the *source* location $s \in M$, the compass direction (bearing) $b \in \{N, NE, NW, E, SE, S, SW, SE, W, None\}$, and a distance d from s in meters. Then we trivially convert to the final target position coordinates. Equation 6.1 now becomes:

$$\hat{y} = position(s, d, b) = f\big(\mathbf{w}, \{o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{wst}, \mathbf{eud}, \mathbf{sen})\}_M\big) \tag{6.2}$$

Due to the complexity of the task, we discuss mainly the classification of the SOURCE block and not the regression error of the TARGET position, similar to Tan and Bansal (2018).

Output Representations As in ROSMI, the model for the Blocks World has three outputs: Source Block, distances, and bearings. We treat each output as a classification sub-task, i.e. predicting one of the k Source Block in the image; the distance between the source and target position in absolute units (regression, 0 to 2); and a bearing label for the direction from source block to target position (classification, 1 out of 9). Some details are lost due to the bearing, which is not a degree but one of the cardinal directions: North, Southwest, etc..

MAPERT's output comprises of two feature vectors, one hidden state for the vision $\mathbf{out_{image}}$ and one hidden state for the language modality $\mathbf{out_w}$ generated by the cross-modality encoder. More specifically, for the bearing predictor, we pass the hidden state $\mathbf{out_{w,0}}$, corresponding to [CLS], to a FF followed by a softmax layer. For predicting the distance, we similarly pass the hidden state $\mathbf{out_{w,0}}$, corresponding to [CLS], to a FF with one output for the distance. Finally, for the source block prediction, we project each of the k image hidden state $\mathbf{out_{image_k}}$ to a single dimension corresponding to the index of the identified source block. We optimise MAPERT by summing the cross-entropy losses for each of the sub-tasks. The final training objective becomes:

$$\mathcal{L} = \mathcal{L}_{source} + \mathcal{L}_{bear} + \mathcal{L}_{dist} \tag{6.3}$$

6.1.2 Results

Ablations on metadata Table 6.1 shows the ablations of the various input features that can be inserted into MAPERT. We pick the combination with the highest SOURCE accuracy and do not display the TARGET error at this stage. The bearing and distance are omitted from this table, since they both fluctuate around 28-33% accuracy and 0.32 mean absolute units error respectively. Thus, we use the SOURCE accuracy as the metric for choosing the right combination of inputs.

MAPERT Ablation	Accuracy (%)	Δ
1. **metaMAPERT**	**49.2**	-
Vision		
2. RoI features	20.0	-29.2
3. RoI features + bounding boxes	28.2	-21.0
4. RoI features + world state	29.0	20.2
Blocks Metadata		
5. Bounding boxes	46.7	-2.5
6. World state	48.0	-1.2
7. World state + Bounding boxes	47.4	-1.8
8. World state + dist edges and corners	**49.2**	0.0
9. World state + NSWE split	45.8	-3.4
10. World state + dist edges and corners + NSWE split	49.0	-0.2
11. Oracle (upper)	100.0	+50.8

Table 6.1: Ablations of the input features for MAPERT. The difference, Delta Δ, is in respect to metaMAPERT (Row 1).

From the ablation, we discover that the best performing features are a mixture of the Blocks World dataset's metadata, the "world state" and the distance between the edges and corners, rather than the vision generated features, which we call metaMAPERT (see row 1 in Table 6.1). The "world state" (see row 6) feature which is the best standalone feature has a decreasing accuracy of only 1.2% from metaMAPERT. The worst performing type of metadata are the Region of Interest (RoI) features, with a considerable decreasing accuracy of -29.2% (row 2). RoI features are visual features generated by an object detector and are compressed representations of pixels from the objects identified in the image. If the objects that are detected look the same as in the blank blocks task, RoIs will look similar, providing more noise than valuable information in the training. It is also observed that input features with low performance, especially the RoI, dramatically affect the rest (see rows 3 and 4, bounding boxes and world state respectively). The rest of the input metadata are spatial coordinates of different kinds and we see the difference in the performance. The bounding boxes (bbox) reach 46.67% (see row 5), twice as much compared to the RoI features (row 1). The world state performs better at 48%, as it adds an extra dimension of height, which the bbox fail to capture. A combination of the world state and the euclidean distance from the edges and corners (row 8) seem to add the most value due to the unique and clear spatial information that we get from the dataset. The "NSWE split", which is another spatial split of the board in four directions, north, south, west, east seem to decrease the accuracy, in both rows 9 and 10, and was not used in the final metaMAPERT, in row 1.

Comparison to state-of-the-art Table 6.2 shows the state-of-the-art (SOTA) results on this version of the dataset. The first model is a simple end-to-end RNN architecture by Bisk et al. (2016b) that could predict the source block and target position as a classification and regression problem, respectively. This model did not perform well in this version of the dataset where the blocks are blank and have no defining features. Tan and Bansal (2018) improved the SOTA to a SOURCE accuracy of 56.1%, with a more complex neural architecture using dual attention and joint training, enabling the model to interpret challenging spatial references. In the most recent paper for this version of the dataset, Mehta and Goldwasser (2019) propose a model that takes in automatically-generated advice with a much simpler architecture. It works best in terms of generating the correct target position and is due to the restrictive advice, where the model is given a general region of the table (i.e. top-right), as an aid. In this last work, both source and target prediction were treated the same way as regression problems, so accuracy was not reported in Table 6.2. Our model metaMAPERT is a large cross-attention neural network that was fully tested on our dataset ROSMI. It is quite flexible in what the input features and output predictions are, but the architecture was specialised for our dataset ROSMI and due to its volume and size can be challenging to train from scratch without any pretraining data. Training MAPERT from scratch and with the main objective to increase the SOURCE classification accuracy, it performs close to the SOTA at 49.16% which is impressive for a complex task as this version of the blocks world. Although not above SOTA, the framework allows for interesting ablations with regards to exploring metadata. We compared metaMAPERT to the SOTA models using a two proportion z-test. We found a significant difference ($z=4.34$, $n=1980$ $p = 0.00001$) between the metaMAPERT and the Dual Attention Expectation Model. Although not above SOTA, the proposed framework allows

Model	**SOURCE**			**TARGET**	
	Accuracy	Median	Mean	Median	Mean
End-to-End RNN (Bisk et al., 2016b)	10.0	3.3	3.5	3.6	3.7
Dual Attention Expectation Model (Tan and Bansal, 2018)	**56.1**	**0.0**	**2.2**	2.8	3.1
4 Regions Restrictive Advice Model (Mehta and Goldwasser, 2019)	NA*	2.2	2.2	**2.2**	**2.2**
metaMAPERT	49.2	0.5	2.4	4.3	4.8

Table 6.2: Final results of MAPERT compared to the previous state-of-the-art for blank blocks. Source block accuracy is the percentage of the time that the model predicted the correct block to move out of the 10 possible blocks in the table. Mean and median are distances in block lengths, the lower is better. *Source block prediction was not treated as a classification problem, so accuracy was not reported.

for interesting ablations with regards to exploring metadata. Our approach is also promising in terms of what can be done with models that can work almost out of the box. The main challenges will shift from modelling to data processing and feature selection.

6.2 Robot Electronic Navigational Chart Instructions (RENCI)

RENCI is a synthetically generated dataset based on ENCs, with generated textual commands for the domain of emergency response with UXV, similar to ROSMI, with the same annotations that are described in Chapter 4, but with ENCs as maps and not OSMs. Due to technical difficulties, RENCI does not come with visual maps as ROSMI does, but only with metadata in the form of text. Despite being synthetic, it can be useful when training neural models on low resource domains. We split RENCI into seven different train/dev splits, to perform a 7-fold cross-validation, and in each fold we use six charts for training and one chart for validation (limiting the validation set to 55 examples as in ROSMI). The "vanilla" MAPERT described in Chapter 5 is not expected to perform well because the landmarks that come with ENC are not as distinguishable as in OSM. Table 6.3 clearly shows the difference between the quality of OSM and ENC metadata, because of the higher distinction between the metadata names in the first one, making landmark grounding a much simpler task in

	ROSMI	RENCI
1.	Bank of America	north san diego harbor anchorage area
2.	IHOP	shelter islands eastern shore anchorage area a 1b
3.	Starbucks	shelter islands eastern shore anchorage area a 1c
4.	Vons 2338	the shelter island yacht basin anchorage area a 1
5.	Saturn Boulevard 0	the shelter island roadstead anchorage area a 1a
6.	Saturn Boulevard 1	the americas cup harbor anchorage area a 2
7.	Conifer Avenue 0	americas cup harbor daybeacon 3
8.	Conifer Avenue 1	americas cup harbor daybeacon 2
9.	Coronado Cays Park	americas cup harbor daybeacon 5
10.	Iowa Street	americas cup harbor daybeacon 4
11.	Cathy Street	americas cup harbor daybeacon 6
12.	Navy Federal Credit Union	americas cup harbor daybeacon 7
13.	Postal Annex	shelter island west end light 4

Table 6.3: Comparison of the first 13 landmarks between ROSMI's and RENCI's first scenario. The same coloured words under each dataset are identical.

OSM than in ENC. Rows 6-12 and 2-6 in RENCI are almost identical with only an identifier phrase or "ID" to show the difference. For a data-driven model, learning to distinguish these landmarks from each other and picking the correct one can be challenging. In ROSMI, the variety is higher, thus making it less challenging for a general-purpose data-driven model to catch. Below, we describe a non-parametric technique that once combined with data-driven models can create very powerful hybrids.

6.2.1 Hybrid-MAPERT for Predicting GPS Locations

The ranking fuction, Best Matching 25 (BM25) (Robertson and Zaragoza, 2009), which is a TF-IDF-like (Term Frequency–Inverse Document Frequency)[2] retrieval function, has been used in prior work (Katsakioris et al., 2022) by the author for performing "fuzzy" search for candidate entities in an EL pipeline for Entity Disambiguation (see Appendix C for the full paper). We draw inspiration from this and apply the BM25 search algorithm to the task of predicting GPS coordinates by "linking" the correct landmark the user mentions in the instruction, to the correct ID of all landmarks on the map.

Hence, we modified our model for map representation and understanding, MAPERT, presented in Chapter 5 and created the hybrid model by adding a non-parametric

[2] http://i.stanford.edu/ ullman/mmds/ch1.pdf

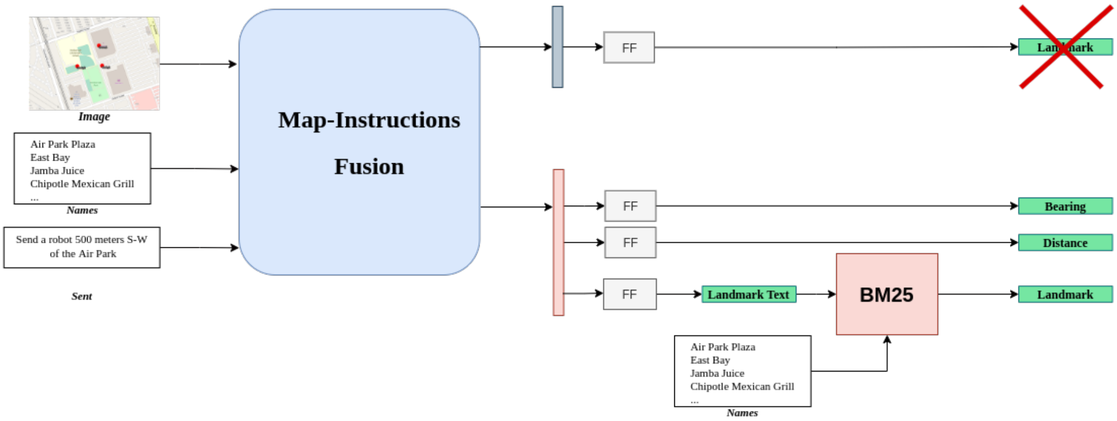


Figure 6.2: Overview of hybrid-MAPERT. The top 'Landmark' prediction head is cut off because hybrid-MAPERT does not use it and instead uses the non-parametric BM25 head below.

module in MAPERT's architecture (see Figure 6.2). MAPERT is a multi-task model, with a task-specific 'head' network for each task that branches off two common, shared hidden layers. This allows for fast and easy modification for task-specific challenges and is one of the greatest advantages of these large data-driven neural models. Instead of using our original head for landmark grounding, meaning directly predicting from the list of landmarks, we append an extra head for the task of finding the name of the landmark in the instruction by tagging the correct tokens in the sentence. For instance, "Send a survey auv vehicle southwest of the **americas cup harbor daybeacon 5**". This new head needs only to "tag" the landmark in the sentence correctly. "Americas cup harbor daybeacon 5" is then passed through the BM25 search, which contains the actual names of our landmarks from the map, to ground the correct GPS landmark coordinate from the list of ENC landmarks. The BM25 is the "non-parametric" part of the model and is not currently trainable, and the landmark token prediction head is trained on the rest of the tasks (bearing, distance) in a multitask learning manner.

6.2.2 Results

MAPERT and Hybrid-MAPERT are both trained and evaluated in 7-cross-validation versions of ROSMI and RENCI, with results shown in Table 6.4. As expected despite MAPERT's success in ROSMI, with an accuracy of 72.1%, MAPERT underperforms

	ROSMI (unseen examples)	**RENCI** (unseen examples)
	Accuracy	Accuracy
Oracle$_{upper}$	100.0	100.0
MAPERT	**72.1**	11.7
Hybrid-MAPERT	54.17	**63.9**

Table 6.4: 7-fold Cross Validations on ROSMI and RENCI using as inputs the sentence instruction and the gold metadata that come with the respective maps, OSM for ROSMI and ENC for RENCI.

in the synthetic dataset of RENCI, with a result of 11.7%. The accuracy is low because: (1) the lack of variability in RENCI, makes it harder for a data-driven model to learn the correct probability distributions of the task; (2) the homogeneity of landmark metadata in RENCI acts as noise rather than valuable input features and makes the task harder. Particularly because of this disadvantage, it is much easier to modify MAPERT than collecting more 'real' data or trying to introduce new features via feature engineering. (3) RENCI comes with no visual features, making the landmark grounding even harder for a model with a specialised visual encoder.

Hybrid-MAPERT which acts as a hybrid solves this problem by relying more in the language encoder than the visual. It performs almost six times better reaching an accuracy of 63.9%, proving our point and solving the issue with the homogeneous metadata. In ROSMI, Hybrid-MAPERT achieves an accuracy of 54.17%, which is also decent for the task at hand and the ease of operation (plug and play). Since ROSMI has higher variability as a dataset and since MAPERT was refined for this dataset, it makes sense that it still surpasses Hybrid-MAPERT[3].

6.3 Discussion

This chapter explored the capabilities of MAPERT, which was introduced in Chapter 5, by keeping the core of the model intact and testing it on two datasets, with varied metadata and new tasks (RQ3). Firstly, Section 6.1 examines MAPERT on the task of the Blocks World (Bisk et al., 2016a), with various input features, both

[3] We perform a two proportion z-test at significance level $\alpha = 0.05$ and population $n = 55$ of the test data (both ROSMI and RENCI), between MAPERT and hybrid-MAPERT accuracy scores. No significant differences between the models on the ROSMI. Hybrid-MAPERT performs significantly better on the RENCI than the regular MAPERT (z= 5.65, $p = 0.0000001, p < \alpha$)

visual and textual, by providing an ablation and also compares them with other SOTA, allowing the exploration of various multi-modal input features and achieving comparable accuracy while keeping the core model intact. Secondly, Section 6.2 makes a direct comparison of the capabilities of MAPERT in RENCI and ROSMI. RENCI is ten times larger than ROSMI but has only twice the amount of unique tokens. Therefore, there is a potential risk for confusion of the models trained on RENCI and the introduction of noise, due to low variance. RENCI is split to perform a 7-fold cross-validation, and in each fold six charts are used for training and one for validation (limiting the validation set to 55 examples as in ROSMI). hybrid-MAPERT is created by modifying the "task" heads and introducing a non-parametric module, necessary to adapt to the new task. We conclude that the task itself can influence the performance of MAPERT, especially on synthetic datasets that might look similar to the dataset the model was trained on. The same applies for the input features that are being used. Having chosen input features that work on one dataset or task, does not imply that it will work as effectively on the other and we see that with RENCI's metadata; even though MAPERT was specialised with metadata, a slight variation changed entirely the performance. Consequently, it is important that MAPERT's core can be kept intact and the rest be adjusted as necessary. This leaves room for continual improvement of MAPERT with just a few modifications when a new task is being introduced or in this case new types of maps. In the next chapter, the possibility of continual improvement of MAPERT is being explored after it has been deployed. This is done by incorporating human interaction in the learning process and by training with new annotated data to improve the performance without re-engineering the model. In addition, we evaluate hybrid-MAPERT as part of a multi-modal collaborative conversational agent with real human experts.

Chapter 7

Expert Evaluation and Continual Learning

In this chapter, we will be reporting on experiments on the maritime domain and will be using charts instead of maps. For this reason, we use hybrid-MAPERT which was introduced in Chapter 6. We create a user interface of the hybrid-MAPERT in the form of a goal-oriented dialogue system that can collaborate with human operators and together plan missions or prevent disasters. Our aim is to (1) evaluate the system with human subjects and (2) coming from the field of Continual and Human In The Loop learning (HITL) (Karmakharm et al., 2019), to answer whether the end-to-end model, i.e. hybrid-MAPERT, can improve performance online by incorporating human interaction in the learning process and by training with new annotated data (RQ4). The chapter is organised as follows.

Section 7.1 presents our interactive learning workflow and experiment set-up. Section 7.2 describes our "chatbot" interface, which combines our understanding model into a conversational agent using the RASA toolkit[1], together with the data augmentation we performed and the evaluation methodology. Section 7.3 presents the results of the online iterative experiment and shows some feedback from the subjects. Finally, Section 7.4 discusses the results and summarises the chapter.

[1] https://rasa.com/docs/

7.1 Method and Experiment Set-up

The experiment was held completely online, due to the COVID-19 pandemic, using Microsoft Office forms and a chatbot interface, called "Hermes", which runs on university servers.

7.1.1 The Task

Subjects were told they were in charge of some robots to rescue vessels in distress. There were two possibilities:

- They received a distress signal, but they need to search an area to locate the vessel in distress (e.g. *'Define search area to cover harbour channel.'*).
- They received a distress signal so they have to localise the vessel in distress on the chart (e.g. *'Send aerial vehicle to search for vessel in distress in the defined search area.'*).

They are controlling three robots:

- Two autonomous ships that can go to the given location to rescue the vessel.
- One drone that can search a large area of the sea to locate the vessel in distress.

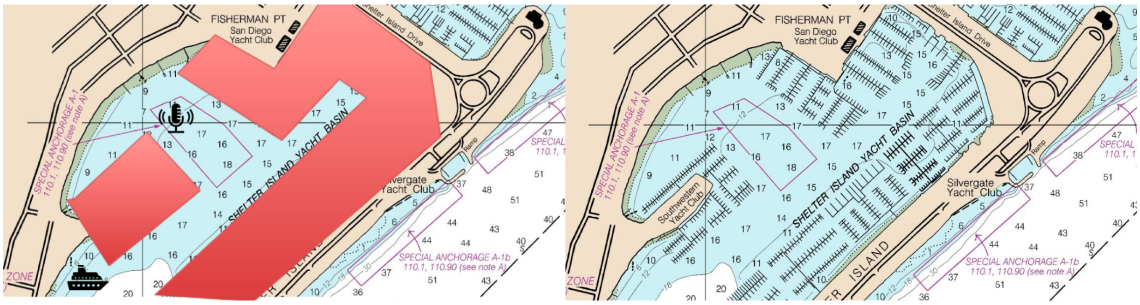
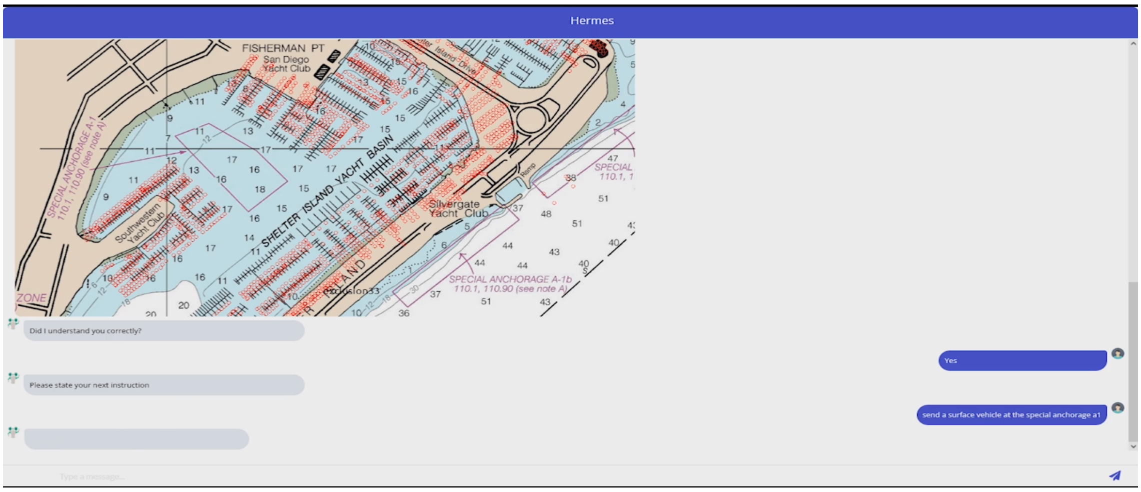

FISHERMAN PT
San Diego Yacht Club
Shelter Island Drive
SHELTER ISLAND YACHT BASIN
Silvergate Yacht Club
Southwestern Yacht Club
SPECIAL ANCHORAGE A-1 110.1, 110.90 (see note A)
SPECIAL ANCHORAGE A-1b 110.1, 110.90 (see note A)
ISLAND
ZONE

Figure 7.1: Both versions of Scenario 1 as provided to the human subjects. Hermes has access to the right version of the chart and the subjects have to collaborate with Hermes to plan a mission as shown on the left version of the chart.

They are provided with two versions of 5 different rescue scenarios in the form of nautical charts, to direct the robots to the location of interest (see Figure 7.1). The first image shows the mission that needs to be performed and the second, which is also "visible" to Hermes, is to help the human subjects identify all the landmarks and refer to them in their instructions. Subjects are prompted to use direction and distance with the help of a compass and the scale of the map which are both provided. Each scenario can have more than one task that needs to be completed and subjects can write as many commands as necessary to successfully plan the mission (see Figure 7.2 for an example interaction). A plan is considered successful if all the required actions are performed. This is checked by the author and qualitatively by each subject (see Section 7.3.1). Examples commands:

- Move Boat 1 30m south from the Main Tower, to board the crew.
- Send Boat 2 450m west north of the Harbour and Search for the vessel.


Hermes
FISHERMAN PT
San Diego Yacht Club
SHELTER ISLAND YACHT BASIN
Silvergate Yacht Club
Southwestern Yacht Club
SPECIAL ANCHORAGE A-1 110.1, 110.90 (see note A)
SPECIAL ANCHORAGE A-1b 110.1, 110.90 (see note A)
ZONE
exclusion33
Type a message...

Did I understand you correctly?

Yes

Please state your next instruction

send a surface vehicle at the special anchorage a1

Figure 7.2: Example interaction between a human subject and Hermes the chatbot on Scenario 1. The red dots symbolise an exclusion zone that has already been communicated between Hermes and the subject.

We collected in total 41 interactions from three subjects across a period of two weeks. Each subject was asked to login daily for a duration of two weeks and interact with Hermes for 15-30 minutes. Unfortunately, due to the subjects' availability the interactions did not happen daily and consistently. However, we did get a reasonable spread of interactions over two weeks. After each interaction, the subjects submitted a form in which they evaluated the interaction and submitted qualitative feedback. Subjects were free to try each scenario as many times as they thought necessary. See Appendix B for the complete Experimental Protocol and detailed pictures of each scenario.

7.1.2 Human-in-the-loop Learning

Our feedback-based learning approach can be used to quickly deploy data-driven interactive models for low-resource domains. It is a simple interactive learning algorithm that deploys a conversational agent, then iteratively improves the parts of the agent which are data-driven using user feedback and selective annotation. A key requirement of this algorithm is the ability to cheaply and efficiently annotate data for chosen user utterances (Iyer et al., 2017). We address this requirement by incorporating an annotating tool inside the interface of our conversational agent, in order for subjects to use it whenever they see fit.

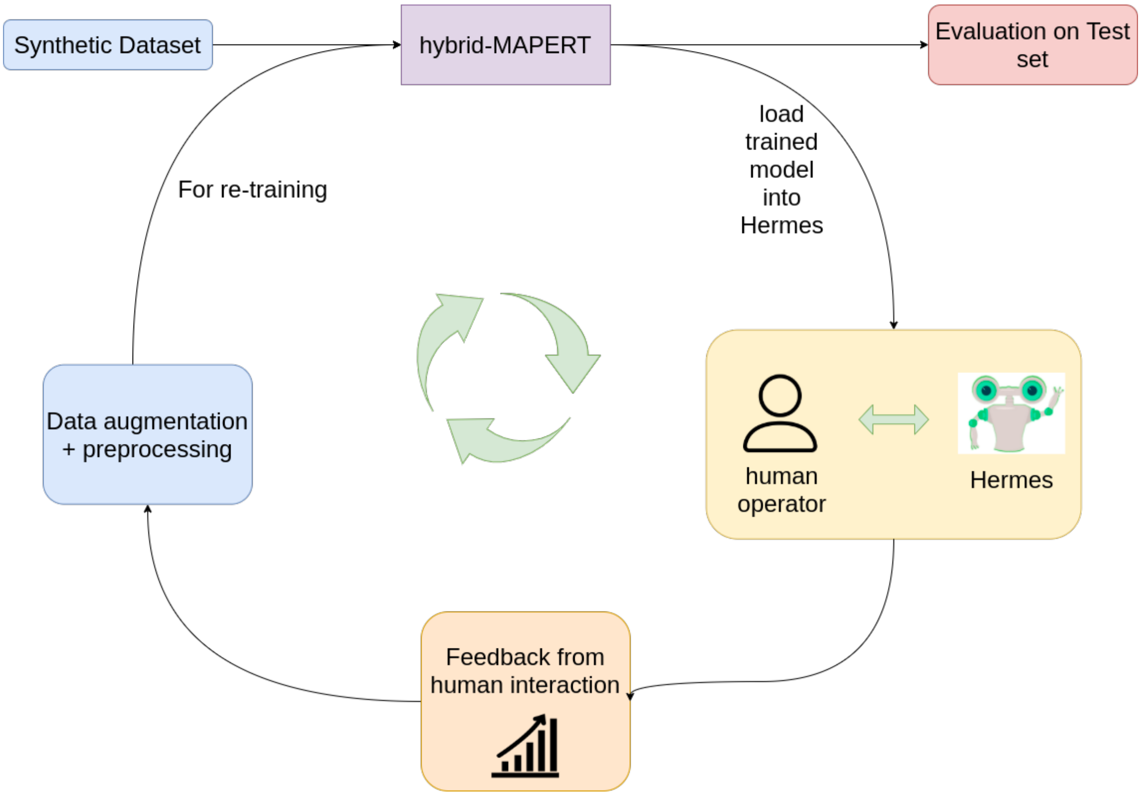

Synthetic Dataset
hybrid-MAPERT
Evaluation on Test set
load trained model into Hermes
For re-training
Data augmentation + preprocessing
human operator
Hermes
Feedback from human interaction

Figure 7.3: The workflow of Human-in-the-loop (HITL) in our study. The human operators provide feedback of the model performance, so as to evaluate it, and annotated data that are later augmented and added to the main dataset for online re-training.

The workflow of our study is seen in Figure 7.3. Our algorithm alternates between stages of training the model and making predictions to gather user feedback, with the goal of improving performance in each successive stage. Hybrid-MAPERT is initially trained on synthetic data S generated by scenario-independent templates (see Chapter 3). Hybrid-MAPERT can predict the goal location by providing the *Landmark*, the *Bearing* and the *Distance*. We extend hybrid-MAPERT with a new classification sub-task to predict the *Object type* of interest in order to display objects that the subjects are moving on the chart. The objects can be one of the following five types: *"survey area", "exclusion zone", "surface vehicle", "aerial vehicle" and "underwater vehicle"*. The trained model is loaded into the chatbot User Interface

"Hermes", which the subjects interact with. After each instruction, Hermes asks the subjects whether the result is "as expected" (see Figure 7.4). Depending on whether the user chooses "Yes/No":

- If the subject chooses "Yes", the subject's language instruction is added to the training set.
- If the subject chooses "No", the algorithm gives the subject the option to annotate the utterance with the correct landmark, bearing, object type and distance before adding it to the training set.
- If the interaction fails due to a technical error (landmark not available in the metadata or task impossible for Hermes to perform), Hermes will ask for the subject to rephrase and repeat.

This procedure can be repeated indefinitely, with the goal to increase hybrid-MAPERT's accuracy; requesting fewer annotations in each successive model iteration. When an interaction is complete, the subject is prompted to a form, so as to provide subjective and qualitative feedback. The data of the interaction are stored and prepared for preprocessing and augmentation before adding them to S, for the next iteration of model training. Appropriate consent was obtained and all experiments going forward were approved by the HWU ethics committee.

7.2 Chatbot Interface

For interfacing with hybrid-MAPERT, a rule-based chatbot, Hermes, is created, using the RASA framework. At a high level, Hermes is a single instruction natural language understanding system, which is trained to understand subject intents and respond accordingly. One of the intents is for calling hybrid-MAPERT, another is to display the predictions to the subjects or to ask subjects for annotations. It is a single instruction system, meaning that it does not keep track of the dialogue state. The goal here is not to create a full-fledged dialogue system but to evaluate and increase the accuracy of hybrid-MAPERT over time with the help of human interaction. In the scope of the emergency response domain for UXV, Hermes is an excellent interface for hybrid-MAPERT.

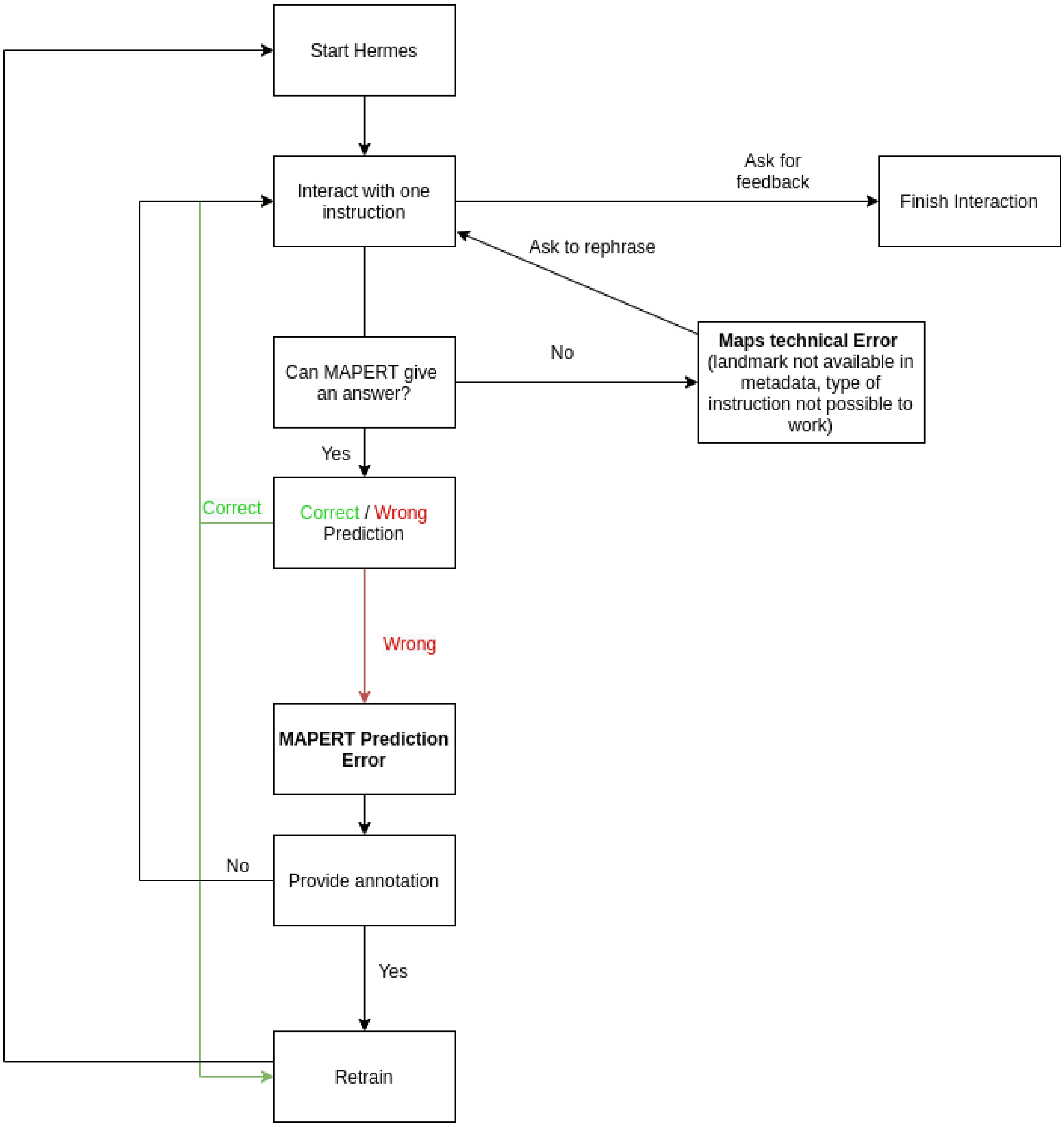

Start Hermes
Interact with one instruction
Ask for feedback
Finish Interaction
Ask to rephrase
Maps technical Error (landmark not available in metadata, type of instruction not possible to work)
No
Can MAPERT give an answer?
Yes
Correct
Correct / Wrong Prediction
Wrong
MAPERT Prediction Error
No
Provide annotation
Yes
Retrain

Figure 7.4: Single instruction procedure.

RASA Framework is an open source machine learning framework for automated text and voice-based conversations. It can understand messages, hold conversations, and connect to messaging channels and APIs. The basic concepts of RASA relevant here are (1) Intents: what the subject is intending to ask about and are stored in the “nlu.yml” as NLU training data, (2) Actions: what action should the bot take upon a specific intent and is stored inside the “domain.yml”, together with all the rule-based responses and entities. In Appendix B, we provide the contents of all the RASA files that were used for creating Hermes. Hybrid-MAPERT and the annotation tool, are part of the RASA framework as an action and they are being toggled by instructions that RASA NLU needed to be trained on. Refer to RASA[2] documentation for more information on creating conversational agents.

Data Augmentation Instead of adding each data point that we collect from the subjects in the dataset S, we modify it so as to increase the amount of data being added in the training loop. The data augmentation strategy to improve the generalisation of hybrid-MAPERT, despite the limited amount of data, is twofold. We either, (1) replace the landmark of the new collected language instruction with five different ones from the scenario at hand (hence creating five examples from one), or (2) keep the landmark of the collected language instruction and generate new sentences using it, with templates collected in the WOz study, described in Chapter 3. Table 7.1, shows the amount of data that S is being augmented with, after each training iteration, together with the number of people and scenarios involved in that stage of collection.

7.2.1 Evaluation Methodologies

Hybrid-MAPERT has been evaluated in a 7-fold cross validation set of RENCI in Chapter 6. Here, a new test set with a mixture of real and synthetic data is formed, for the sole purpose of having a quantitative evaluation method between the iterations of the experiment. The test set consists of 66 annotated examples that were carefully chosen by the author and an industry professional in order to be as diverse as possible and as a measure of quality with excessive difficulty. With the four outputs of hybrid-MAPERT, *landmark, distance, object_type and bearing*, the destination location is indirectly predicted. Success is measured by the model's prediction capabilities on all four outputs and is seen as "Total Accuracy" in Figure 7.5, together with a separate accuracy for each output; "Landmark Accuracy" for landmark prediction; "Bearing Accuracy" for bearing prediction and "Object Type Accuracy" for predicting the type of object the subjects want to add on the map.

In addition, subjective and qualitative feedback is gathered. Subjects after each interaction were asked to (1) classify the interaction as successful or not, (2) measure

[2] https://rasa.com/docs/rasa/playground

the quality of the interaction using a 5-point Likert scale, with 1 meaning Very Low Quality and 5 Very High Quality, and (3) an open question of what they liked or not about Hermes.

7.3 Results & Further Analysis

Table 7.1 shows the accuracy of every training iteration, accompanied by other statistics for the model iteration. The first model, trained purely on S on the 4th of Oct 2021, has a total accuracy of 34.85% indicating the complexity of the task even though the test set is partially synthetic. We see the highest accuracy jump on the 13th of Oct 2021 (3rd iteration of training) to 40.91%, potentially due to the high amount of data collected and the high variance from the participation of all the subjects and across all scenarios. On the 4th iteration despite the data that were added being only around 200, we see a further increase to 42.42%. This increase might be associated with the quality of the data collected, which span across almost all scenarios (except scenario 4). We see a small drop at the last iteration that could be due to the addition of noise or overfitting; in the last 3 iterations, data was collected only from one subject at a time.

Model	**# examples**	**# people**	**Iteration**	**scenarios**	**Accuracy (%)**
Oracle	-	-	-	-	84.85
mapert_04/10	4954	0	0	-	34.85
mapert_07/10	5014	2	1	1	34.85
mapert_13/10	5704	3	2	1,3,4,6,7	40.91
mapert_14/10	5929	1	3	1,3,6,7	**42.42**
mapert_15/10	6139	1	4	4,6,7	**42.42**
mapert_20/10	6244	1	5	6,7	40.91

Table 7.1: Models and dataset statistics during the human evaluation. The second part of the model names represent the dates of training. The number of examples is the total set of data per iteration.

For comparison purposes, Table 7.2 shows an offline iterative training of the model, on data that are 100% synthetically generated. We test on various training data sizes, so as to show the value of human participants in the process of online learning, by comparing model iterations from our HITL framework on Table 7.1 to an offline synthetic data training process without any human intervention. The best offline performing model is mapert_3 has an accuracy of 39.39%, around 3% less than the best performing model of the HITL online training. This shows the importance of human intervention when generating new data, even if most of the data are synthetically generated. The model trained on 9135 examples, around 3000 more synthetic examples than the best performing HITL model is on Table 7.1, performs even worse with an accuracy of 36.36%. All models trained on fully synthetic data performed worse than the best HITL model.

Model	**# examples**	**Accuracy (%)**
Oracle	-	84.85
mapert_2	4954	39.39
mapert_3	5704	39.39
mapert_4	6244	37.88
mapert_5	9135	36.36

Table 7.2: Models trained only on synthetically generated data.

In Figure 7.5, we see a more detailed graph of all the accuracies across all model iterations. On the 3rd iteration, the task that gains the most from the HITL training is the object type prediction, with "Object Type Accuracy" going from 45.45% to 53.01%, potentially due to the fact that it is a new task that was added to hybrid-MAPERT and there is room for improvement. In contrast, the "Bearing Accuracy", which is a task from both RENCI and ROSMI, meaning that the synthetic data have already a lot of variability, is initially high at 98.48% with not much room to expand. In addition, the already high-performing "Landmark Accuracy" increased from 90.91% to 93.94%.

7.3.1 Subjective & Qualitative Feedback

Table 7.3 shows which scenarios were marked successful and by which subject by the end of the experiment. Scenarios 3 and 7 were successful for subject "aaa76" from the beginning and scenarios 1, 3 and 6 for subject "ccc12". Subject "ccc12"'s immediate success on these three scenarios, from the first try, might be because the interaction happened after the 3rd iteration of training and as shown from Table 7.1, the 3rd iteration has the highest increase in accuracy. Subject "ccc12" started with scenario 7 and then with scenario 4, and similarly to the rest, failed to successfully complete them on the first try. No one successfully finished scenario 4, indicating the complexity of the scenario and potential technical issues that we will discuss

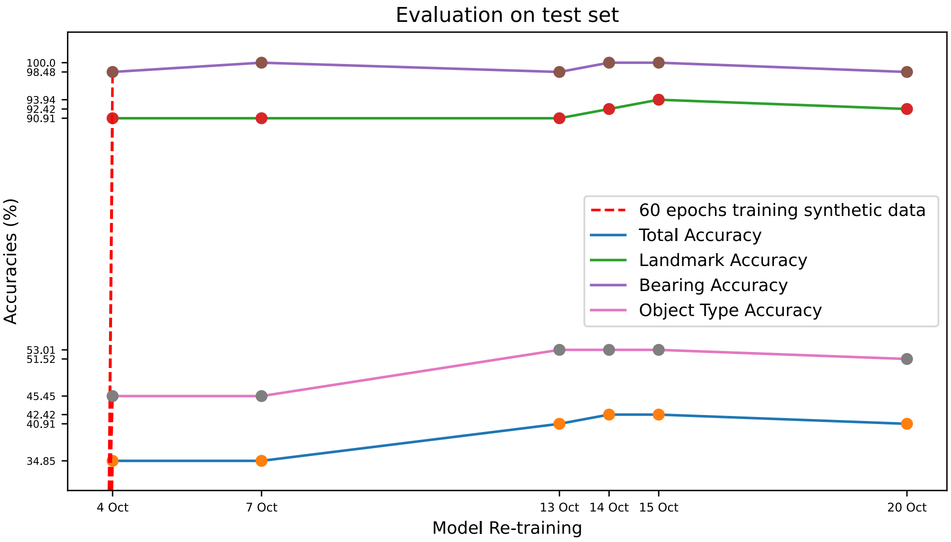


Figure 7.5: Accuracy on the test set throughout all iterations.

further below.

Figure 7.6 shows the median subject ratings for each model iteration from Table 7.1. Consistently, in Figure 7.6 the highest jump happens between the first and second iteration of the model, from a median of 2.0 to 4.0.

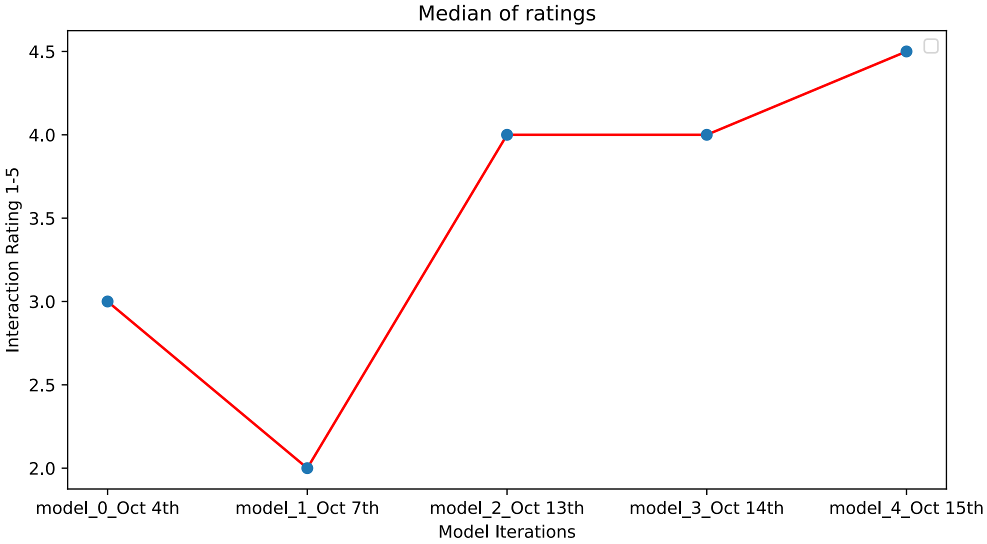


Figure 7.6: Median subject ratings of every model iteration, on a scale from 1 to 5. We have no rating for the final model that was trained on Oct 20th.

Based on the open question the subjects were asked, an inductive, thematic analysis was done by the author using grounded theory with open coding from Strauss (1987). Themes identified include:

Scenarios	**aaa76**	**bbb22**	**ccc12**
scenario 1:			
-First	NO	NO	YES
-Last	YES	YES	YES
scenario 3:			
-First	YES	NO	YES
-Last	YES	NO	YES
scenario 4:			
-First	NO	NO	NO
-Last	NO	NO	NO
scenario 6:			
-First	NO	NO	YES
-Last	YES	NO	YES
scenario 7:			
-First	YES	NO	NO
-Last	YES	YES	YES

Table 7.3: Classification of first and last interaction of each scenario as successful or not, by each subject.

Theme 1 Visual aids and enhancements: Despite the scale that was given for each scenario, subjects had trouble calculating distances without an online visual aid, on the chart they were meant to perform the task. There were also comments on enhancing the way "exclusion zones" appeared on the chart, as it was not always possible to cover the desired landmark, even though the predictions of hybrid-MAPERT were correct. In addition, comments on the size and the ability to modify the size of "survey areas" were made, as it caused some frustration to their interaction as well.

Theme 2 Landmark mismatches: There was a disconnect between some landmarks that were visible on the chart and the actually ones available as metadata (especially scenario 4). This caused some frustration. A cheat sheet was given with the available metadata landmarks, but some subjects could not find them on the chart.

7.4 Discussion and Takeaways

From the evaluation results, both quantitative and qualitative, we saw the potential of improving a model continually from expert subject feedback. Further questions occur, which provide opportunities for more exploration and improvement, and are covered here. Firstly, it would be interesting to try the same experiment with two different groups of testers, one who gets a trained model and one that does not get a trained model, to see how much their judgement of the model is affected by their adaptation to the capabilities of the program. A question arises on how much does the belief of the subjects, that they are working with a model that is retrained and continually improved, affect the overall qualitative feedback and evaluation of the system. Secondly, the subjects that interacted with the model were also annotating the examples. This methodology might be useful when the availability of more testers is not possible but it is prone to more errors due to mental fatigue of the testers. One subject specifically mentioned that they accidentally put wrong annotations in two cases. This might introduce unnecessary noise in the data. Having extra "trained" annotators can play an important role in this type of online learning. Last, but not least, variability in the scenarios and subjects, aids the generalisability of the model and makes it less prone to human errors.

This chapter is concerned with the improvement of a data-driven system over time, based on human interaction and feedback in the form of annotated data with minimal intervention. The model described here is automatically retrained overnight with the expert data collected and augmented the same day. As with any approach it has its drawbacks, such as the potential noise that may incur, due to the subjects' annotations. This complete feedback loop, without intermediate intervention, opens up new ways of building interactive systems that can function almost from scratch and get better in time without the need for large amounts of expensive expert data or model engineering. In addition, we got the chance to evaluate with human subjects our map understanding model introduced in previous chapters and get valuable feedback for future directions and functionality amplifications.

Chapter 8

Conclusions and Future Directions

This thesis described the development and evaluation of data-driven and non-parametric methods for mapping natural language instructions on a map/chart, using low-resource expert data.

This chapter summarises the main contributions and findings in Section 8.1 and indicates possible avenues for future work in Section 8.2 (see Table 8.1 for a summary).

8.1 Contributions and Findings

This section is broken down into four subsections with alignment to the research questions presented in the introductory chapter.

RQ1: How to address the challenge of data scarcity on low-resource expert domains, such as when working with Autonomous Underwater Vehicles (AUV) (Chapters 3 and 4)?
Experts, in first responder scenarios, off-shore energy installations, defence or search and rescue, typically create a plan for vehicles using a visual interface on dedicated hardware on-shore, days before the mission. This planning process is complicated and requires expert knowledge. In situated dialogue, each user can perceive the environment in a different way, meaning that referring expressions need to be carefully selected and verified, especially if the shared environment is ambiguous (Fang et al., 2013). In order to investigate the way human experts would collaboratively make

plans, via a CA for remote UXV, Chapter 3 presented a two-wizard WoZ study for collecting data on a collaborative task, identifying the importance of mixed modalities and object referencing for successful interaction during mission planning. The objective is to allow more precise and efficient plans and reduce operator training time. In addition, it will allow for anywhere access to planning for in-situ replanning in fast-moving dynamic scenarios.

Data analysis shows that multi-modality is key to successful interaction, measured both quantitatively and qualitatively via user feedback. With the collected WoZ data, the main strategies of how to plan a mission and make data-driven simulations possible, are captured and used to create a synthetic dataset, *Robot Electronic Navigational Chart Instructions (RENCI)*, described in Chapter 4.

In addition, Chapter 4 presented a novel challenge, *Robot Open Street Map Instructions (ROSMI)*, which refers to the task of understanding maps and process instructions, in order to generate GPS locations for the UXV. The ROSMI corpus is to aid in the advancement of state-of-the-art visual-dialogue tasks, including reference resolution and robot-instruction understanding. The domain described here concerns robots and autonomous systems being used for inspection and emergency response. The ROSMI corpus is unique in that it captures interaction grounded in map-based visual stimuli that is both human-readable to non-experts but also contains rich metadata that is needed to plan and deploy robots and autonomous systems, thus facilitating human-robot teaming. Furthermore, RENCI a synthetically generated dataset based on ENCs that is similar but 10 times larger than ROSMI, was also created. Despite being synthetic, this dataset can be used to bootstrap neural model training on low-resource domains. In addition, rather than focusing solely on neural model development, we hope to facilitate more research into data production and processing (Zhao et al., 2020; Li et al., 2021; Shen et al., 2021).

RQ2: How do we develop an end-to-end neural model that understands natural language instructions referring to entities on maps/charts, i.e. that can read maps (Chapter 5)?

Robust situated dialogue requires the ability to process instructions based on spatial information, which may or may not be available. In Chapter 5, a novel end-to-end method is proposed, Map Encoder Representations from Transformers (MAPERT), a model based on LXMERT that can extract spatial information from text instructions and attend to landmarks on maps (OSM) referred to in a natural language instruction. Whilst, OSM is a valuable resource, as with any open-sourced data, there is noise and variation in the names referred to on the map, as well as, variation in natural language instructions, hence the need for data-driven methods over rule-based systems. The error analysis, in Section 5.3, shows that if no metadata is available then the model can use Vision, although this is clearly a harder task. We showed that Vision does seem to help in examples with complex spatial relations (i.e. 'near', 'next to') or ambiguity between entities.

RQ3: Is the end-to-end model, developed in (2), task-specific or does it generalise to new tasks (Chapter 6)?

Chapter 6 explores whether or not the end-to-end model (MAPERT), developed in Chapter 5, is generalisable to new tasks and datasets. MAPERT's core is a large, cross-modality encoder with multiple, task-specific prediction "heads". By keeping the core of the model intact, it can be easily adapted to any multi-modal task.

Firstly, it was tested on a different task of moving blocks around on a series of images, with varied input features, both visual and textual. A completely different task, with no GPS locations as predictions but with spatial relations and referring expressions. In this context, MAPERT despite the different task it achieved comparable accuracy with the previous SOTA with a few inexpensive changes in the model's output, while keeping the core model intact.

Secondly, MAPERT is tested on a similar task "RENCI" (see Chapter 4), with GPS location prediction but with input features of different variation and natural language. Despite the similarities in the task, due to the type of input features, adapting the output heads with an extra non-trainable prediction output was necessary to achieve desirable performance, thus creating hybrid-MAPERT with the core of the model remaining still intact.

From the aforementioned experiments, it is concluded that MAPERT can be used for new tasks and with varying inputs allowing the fast bootstrapping of neural model research, due to the fact that the core of the model can remain intact and alleviate expensive modifications. With the right modifications depending on the task itself and the input features MAPERT can achieve the desirable performance, particularly impressive for using data-driven models in low-resource expert domains.

RQ4: How do we embed the model in (2) into a Human-In-the-loop framework for continual improvement through short interactions with experts (Chapter 7)?

Previously, MAPERT and hybrid-MAPERT have been evaluated on static test sets on various datasets and tasks. Hybrid-MAPERT, due to the maritime nature of the task and the capabilities of this version of MAPERT on ENC charts, is inserted in a goal-oriented multi-modal dialogue interface in order to plan for UXV missions collaboratively with human operators. In Chapter 7, hybrid-MAPERT was evaluated with real human subjects, so as to proof-test its capabilities and to get valuable feedback for future directions. The dialogue system acted as a HITL interface and was used to gather data, so as to retrain the model overnight from real human subjects. The model successfully showed improvement compared to the offline iterative training. This entire feedback loop, which requires no intermediate expert intervention, opens up new ways of developing interactive systems that by bootstrapping them with synthetic data, can work nearly from scratch and improve over time without the use of significant amounts of expensive expert data or model engineering.

8.2 Future Work

Over the course of this thesis, multiple questions were raised, however, the work can be extended in various ways, which are briefly described below.

On creating expert data. Next steps for tackling the data scarcity problem could include:

1. Training a Reinforcement Learning agent on simulated dialogues that are fully data-driven with the reward function being derived from subjects' preferences, optimising for plan quality and speed. Moreover, supervised approaches that require less data to learn, such as the Hybrid Code Networks (HCN) (Williams et al., 2017), could be used for the creation of such a CA.
2. A standalone module for grounding referring expressions using metadata, since a standalone module could be more easily adapted to other similar domains

and UXV, as plug and play “API”.

Accurate representation of structured environments. In this thesis, electronic map understanding and vision-language grounding is explored, in Chapters 4 and 5. Further work involves exploring the capabilities of the end-to-end model developed here, MAPERT, on more ambiguous instructions with emphasis on spatial relations, such as *between*, *near*, *next to* etc., and on metadata that are inaccurate and new tasks (different types of maps/charts, i.e. ENC). Finally, as standalone modules, large attention models similar to MAPERT can be used in an end-to-end dialogue system for remote robot planning, whereby multi-turn interaction can handle ambiguity and ensure reliable and safe destination prediction before instructing remote operations.

On more generalisable multi-modal collaborative CAs. Chapter 6 showed how a data-driven black box model can be shared throughout tasks, with different input and output features. Modifications are still needed, but compared to expensive re-engineering of data-driven models, feature and data engineering seems a more cost and time effective strategy to move forward. Based on these outcomes, new questions arise:

- on whether MAPERT can keep improving continually on different tasks, after its deployment by incorporating human feedback in the learning process.
- how MAPERT will perform with real experts in a multi-modal collaborative interaction.

On continual learning. The complete feedback learning loop from human interactions, that is introduced in Chapter 7, opens up new ways of building interactive systems. Future directions on this type of learning would be to experiment on larger scale and for longer periods of time with numerous human testers, exploring the role of human psychology in these type of CAs (with more than one group of testers) and the effect of human mental fatigue on the quality of the data collected.

8.3 Conclusions

This chapter summarised the work presented in this thesis and discussed the contributions made. Finally, it presented directions for future work. In addition, we present Table 8.1 by way of a final summarisation of the work.

	Research Question	**Finding**	**Future Directions**
RQ1 (Ch.3&4)	How to address the challenge of data scarcity on low-resource expert domains, such as when working with Autonomous Underwater Vehicles (AUV)?	Identified the importance of object referencing on the emergency response domain for successful interaction during mission planning. User feedback showed that multimodality is key to successful interaction. Findings include three datasets, one from a Wizard of Oz study, one based on maps (ROSMI) and one based on charts (RENCI).	• Reinforcement Learning agent built on simulated dialogues from the WOz data collected. • A grounding module using metadata from the shared environment of operations, allowing easy generalisation and scaling.
RQ2 (Ch.5)	How do we develop an end-to-end neural model that understands natural language instructions referring to entities on maps/charts, i.e. that can read maps?	Findings include a novel, end-to-end, deep learning model (MAPERT) that enables multi-modal fusion of features, is capable of processing situated language instructions and ground them on different environments, such as OSM maps, using various types of input data, and can do so for maps and charts that it has not seen before.	• Exploring this further by training the model on these type of instructions and on metadata that are scarce and inaccurate. • Generalisability of the understanding module on new tasks and datasets.
RQ3 (Ch.6)	Is the end-to-end model, developed in (2), task-specific or does it generalise to new tasks ?	Findings are that the task itself and the input features can greatly affect the performance of the model, regardless of the dataset, but with the core of MAPERT intact and minor modifications we can achieve desirable performance in new tasks or domains (hybrid-MAPERT).	• MAPERT can be adapted to the new task with minor changes but can it be improved continually? • How does it respond with real human experts?
RQ4 (Ch.7)	How do we embed the model in (2) into a Human-In-the-loop framework for continual improvement through short interactions with experts?	The model in (2) can be improved over time using the HITL framework with synthetic annotated data without the intervention of an engineer.	• Explore the effect of the human psychology in the experiment with two groups of testers. • Try a bigger scale experiment and longer to see whether or not noise will affect the quality of the data overtime.

Table 8.1: Research questions, discovered findings, and the future work of this thesis.

Bibliography

Ai, H. and Lu, X. (2010). A web-based system for automatic measurement of lexical complexity. *In Proceedings of the 27th Annual Symposium of the Computer-Assisted Language Consortium (CALICO-10).*

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. S., and Weinert, R. (1991). The HCRC Map Task Corpus. *Language and Speech*, 34(4):351–366.

Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and van den Hengel, A. (2018). Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3674–3683, Los Alamitos, CA, USA. IEEE Computer Society.

Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2(4):319–342.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (2015). VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV).*

Arslan, D. and Eryiğit, G. (2021). Evaluation of wizard-of-oz and self-play data collection techniques for turkish goal-oriented dialogue agents. In *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6.

Austin, J. L. (1962). *How To Do Things With Words.* Oxford University Press.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

Bastianelli, E., Croce, D., Vanzo, A., Basili, R., and Nardi, D. (2016). A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2747–2753. AAAI Press.

Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., Pinto, N., and Turian, J. (2020). Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online. Association for Computational Linguistics.

Bisk, Y., Marcu, D., and Wong, W. (2016a). Towards a dataset for human computer communication via grounded language acquisition. In *Proceedings of the AAAI'16 Workshop on Symbiotic Cognitive Systems*, volume WS-16-01 -, pages 729–732.

Bisk, Y., Shih, K., Choi, Y., and Marcu, D. (2018a). Learning interpretable spatial operations in a rich 3d blocks world. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

Bisk, Y., Shih, K., Choi, Y., and Marcu, D. (2018b). Learning interpretable spatial operations in a rich 3d blocks world. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

Bisk, Y., Yuret, D., and Marcu, D. (2016b). 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*, pages 751–761, San Diego, California. Association for Computational Linguistics.

Blaylock, N. (2011). Semantic annotation of street-level geospatial entities. In *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing*, pages 444–448.

Blazhenkova, O. and Kozhevnikov, M. (2009). The new object-spatial-verbal cognitive style model: Theory and measurement. *Applied Cognitive Psychology*, 23(5):638–663.

Bloom, P., Garrett, M. F., Nadel, L., and Peterson, M. A. (1999). Perspective Taking and Ellipsis in Spatial Descriptions. In *Language and Space*. The MIT Press.

Blukis, V., Brukhim, N., Bennett, A., Knepper, R. A., and Artzi, Y. (2018). Following high-level navigation instructions on a simulated quadcopter with imitation learning. In *Robotics: Science and Systems (RSS)*.

Blukis, V., Knepper, R. A., and Artzi, Y. (2020). Few-shot object grounding and mapping for natural language robot instruction following. In *CoRL*.

Blukis, V., Terme, Y., Niklasson, E., Knepper, R. A., and Artzi, Y. (2019). Learning to map natural language instructions to physical quadcopter control using simulated flight. In *Proceedings of the Conference on Robot Learning*.

Bonial, C., Marge, M., Artstein, R., Foots, A., Gervits, F., Hayes, C. J., Henry, C., Hill, S. G., Leuski, A., Lukin, S. M., Moolchandani, P., Pollard, K. A., Traum, D. R., and Voss, C. R. (2017). Laying down the yellow brick road: Development of a wizard-of-oz interface for collecting human-robot dialogue. *CoRR*, abs/1710.06406.

Boye, J., Fredriksson, M., Götze, J., Gustafson, J., and Königsmann, J. (2014). Walk this way: Spatial grounding for city exploration. In Mariani, J., Rosset, S., Garnier-Rizet, M., and Devillers, L., editors, *Natural Interaction with Robots, Knowbots and Smartphones*, pages 59–67, New York, NY. Springer New York.

Brébisson, A. d., Simon, É., Auvolat, A., Vincent, P., and Bengio, Y. (2015). Artificial neural networks applied to taxi destination prediction. *CoRR*, abs/1508.00021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Brutzkus, A. and Globerson, A. (2019). Why do larger models generalize better? A theoretical perspective via the XOR problem. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 822–830. PMLR.

Budzianowski, P. and Vulić, I. (2019). Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Butler, C., Oster, H., and Togelius, J. (2020). *Human-in-the-Loop AI for Analysis of Free Response Facial Expression Label Sets.* Association for Computing Machinery, New York, NY, USA.

Byron, D., Koller, A., Striegnitz, K., Cassell, J., Dale, R., Moore, J., and Oberlander, J. (2009). Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 165–173, Athens, Greece. Association for Computational Linguistics.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, Cham. Springer International Publishing.

Chai, J. Y., Fang, R., Liu, C., and She, L. (2016). Collaborative language grounding toward situated human-robot dialogue. *AI Magazine*, 37:32–45.

Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. (2018). Gated-attention architectures for task-oriented language grounding. 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. AAAI Press.

Chen, D. L. and Mooney, R. J. (2011). Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11. AAAI Press.

Chen, H., Liu, X., Yin, D., and Tang, J. (2017). A survey on dialogue systems: Recent advances and new frontiers. *CoRR*, abs/1711.01731.

Chen, H., Suhr, A., Misra, D., Snavely, N., and Artzi, Y. (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, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., and Gao, W. (2021). Pre-trained image processing transformer.

Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2020). Uniter: Universal image-text representation learning. In *ECCV*.

Chi, T., Shen, M., Eric, M., Kim, S., and Hakkani-Tür, D. (2020). Just ask: An interactive learning framework for vision and language navigation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2459–2466. AAAI Press.

Cirik, V., Morency, L.-P., and Berg-Kirkpatrick, T. (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), pages 781–787, New Orleans, Louisiana. Association for Computational Linguistics.

Clark, H. H. and Brennan, S. (1991). Grounding in communication. In *Perspectives on socially shared cognition.*

Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Corbin, J. and Strauss, A. (2008). *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage.

Dagan, I. and Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In Prieditis, A. and Russell, S., editors, *Machine Learning Proceedings 1995*, pages 150–157. Morgan Kaufmann, San Francisco (CA).

Dahlback, N. and Jonsson, A. (1989). Empirical studies of discourse representations for natural language interfaces. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England. Association for Computational Linguistics.

Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M. F., Parikh, D., and Batra, D. (2017). Visual dialog. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1080–1089.

Deng, J., Socher, R., Fei-Fei, L., Dong, W., Li, K., and Li, L.-J. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, volume 00, pages 248–255.

DeVault, D., Oved, I., and Stone, M. (2006). Societal grounding is essential to meaningful language use. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 747–754. AAAI Press.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dickinson, J., Byblow, W. D., and Ryan, L. (1993). Order effects and the weighting process in workload assessment. *Applied Ergonomics*, 24(5):357 – 361.

Ehsan, U., Harrison, B., Chan, L., and Riedl, M. O. (2017). Rationalization: A neural machine translation approach to generating natural language explanations.

Elahi, M., Ricci, F., and Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Comput. Sci. Rev.*, 20(C):29–50.

Eric, M., Krishnan, L., Charette, F., and Manning, C. D. (2017). Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Eshghi, A., Hough, J., and Kempson, R. (2012). Conversational interactions: Capturing dialogue dynamics.

Eshghi, A., Purver, M., and Hough, J. (2011). Dylan: Parser for dynamic syntax. *Technical report, Queen Mary University of London.*

Everingham, M., Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vision*, 88(2):303–338.

Fang, R., Liu, C., She, L., and Chai, J. (2013). Towards situated dialogue: Revisiting referring expression generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 392–402. Association for Computational Linguistics (ACL).

Franklin, N., Tversky, B., and Coon, V. (1992). Switching points of view in spatial mental models. *Memory Cognition*, 20:507–518.

Fu, J., Korattikara, A., Levine, S., and Guadarrama, S. (2019). From language to goals: Inverse reinforcement learning for vision-based instruction following. In *International Conference on Learning Representations*.

Gao, H., Shou, Z., Zareian, A., Zhang, H., and Chang, S.-F. (2018). Low-shot learning via covariance-preserving adversarial augmentation networks. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 983–993, USA. Curran Associates Inc.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Götze, J. and Boye, J. (2016). SpaceRef: A corpus of street-level geographic descriptions. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3822–3827, Portorož, Slovenia. European Language Resources Association (ELRA).

Gou, Y., Lei, Y., and Liu, L. (2021). Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Haas, C. and Riezler, S. (2016). A corpus and semantic parser for multilingual natural language querying of OpenStreetMap. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 740–750, San Diego, California. Association for Computational Linguistics.

Haihong, E., Zhang, W., and Song, M. (2019). Kb-transformer: Incorporating knowledge into end-to-end task-oriented dialog systems. *2019 15th International Conference on Semantics, Knowledge and Grids (SKG)*, pages 44–48.

Haklay, M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18.

Hancock, B., Bordes, A., Mazare, P.-E., and Weston, J. (2019). Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.

Harnad, S. (1990). The symbol grounding problem. *Phys. D*, 42(1–3):335–346.

Hastie, H., Garcia, F. J. C., Robb, D. A., Patron, P., and Laskov, A. (2017a). Miriam: a multimodal chat-based interface for autonomous systems. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction, ICMI'17*, pages 495–496, NY, USA. ACM.

Hastie, H., Garcia, F. J. C., Robb, D. A., Patron, P., and Laskov, A. (2017b). Miriam: A multimodal chat-based interface for autonomous systems. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, page 495–496, New York, NY, USA. Association for Computing Machinery.

Hastie, H., Lohan, K., Chantler, M., Robb, D. A., Ramamoorthy, S., Petrick, R., Vijayakumar, S., and Lane, D. (2018a). The ORCA hub: Explainable offshore robotics through intelligent interfaces. In *Proc. of Explainable Robotic Systems Workshop, ACM HRI Conference*, pages 1–2.

Hastie, H., Lohan, K., Mike, C., David A., R., Subramanian, R., Ron, P., Sethu, V., and David, L. (2018b). The orca hub: Explainable offshore robotics through intelligent interfaces. In *Proceedings of the HRI Workshop on Explainable Robotic-Systems*, HRI '18, New York, NY, USA. Association for Computing Machinery.

Hastie, H., Robb, D. A., Lopes, J., Ahmad, M., Le Bras, P., Liu, X., Petrick, R. P., Lohan, K., and Chantler, M. J. (2019). Challenges in collaborative hri for remote robot teams. In *Proceedings of the CHI2019 Workshop on The Challenges of Working on Social Robots th at Collaborate with People (SIRCHI2019).*

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hemachandra, S., Walter, M. R., Tellex, S., and Teller, S. (2014). Learning spatial-semantic representations from natural language descriptions and scene classifications. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2623–2630.

Hentschel, M. and Wagner, B. (2010). Autonomous robot navigation based on openstreetmap geodata. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, pages 1645–1650.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Howes, C., Healey, P., and Mills, G. (2009). A: An experimental investigation into... B: ...split utterances. In *Proceedings of the SIGDIAL 2009 Conference*, pages 79–86, London, UK. Association for Computational Linguistics.

Huang, B., Bayazit, D., Ullman, D., Gopalan, N., and Tellex, S. (2019). Flight, Camera, Action! Using Natural Language and Mixed Reality to Control a Drone. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Ilinykh, N., Zarrieß, S., and Schlangen, D. (2019). Tell me more: A dataset of visual scene description sequences. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 152–157, Tokyo, Japan. Association for Computational Linguistics.

Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J., and Zettlemoyer, L. (2017). Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Vancouver, Canada. Association for Computational Linguistics.

Johansson, V. (2008). Lexical diversity and lexical density in speech and writing: a developmental perspective. Technical report, Lund University, Dept. of Linguistics and Phonetics.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition.

Karmakharm, T., Aletras, N., and Bontcheva, K. (2019). Journalist-in-the-loop: Continuous learning as a service for rumour analysis. 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): System Demonstrations*, pages 115–120, Hong Kong, China. Association for Computational Linguistics.

Katsakioris, M. M., Konstas, I., Mignotte, P. Y., and Hastie, H. (2020). *ROSMI: A Multimodal Corpus for Map-Based Instruction-Giving*, page 680–684. Association for Computing Machinery, New York, NY, USA.

Katsakioris, M. M., Konstas, I., Mignotte, P. Y., and Hastie, H. (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*, pages 11–21, Online. Association for Computational Linguistics.

Katsakioris, M. M., Laskov, A., Konstas, I., and Hastie, H. (2019). Corpus of multimodal interaction for collaborative planning. In *Proceedings of the SpLU-RoboNLP 2019 Workshop in conjunction with the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*.

Katsakioris, M. M., Zhou, Y., and Masato, D. (2022). Entity linking in tabular data needs the right attention.

Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. (2014a). ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar. Association for Computational Linguistics.

Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. L. (2014b). Referit game: Referring to objects in photographs of natural scenes. In *Proceedings of EMNLP*.

Kelleher, J. D. and Kruijff, G.-J. M. (2006). Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 1041–1048, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Jaśkowski, W. (2016). ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games*, pages 341–348, Santorini, Greece. IEEE. The best paper award.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17.

Kirstain, Y., Lewis, P., Riedel, S., and Levy, O. (2021). A few more examples may be worth billions of parameters. *ArXiv*, abs/2110.04374.

Kiselev, A. and Loutfi, A. (2012). Using a mental workload index as a measure of usability of a user interface for social robotic telepresence. *2nd Workshop of Social Robotic Telepresence in Conjunction with IEEE International Symposium on Robot and Human Interactive Communication 2012*.

Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., and Zhai, X. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Kollar, T., Tellex, S., Roy, D., and Roy, N. (2010). Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, HRI '10, pages 259–266, Piscataway, NJ, USA. IEEE Press.

Kolve, E., Mottaghi, R., Gordon, D., Zhu, Y., Gupta, A., and Farhadi, A. (2017). AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474.

Kreutzer, J., Riezler, S., and Lawrence, C. (2021). Offline reinforcement learning from human feedback in real-world sequence-to-sequence tasks. In *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*, pages 37–43, Online. Association for Computational Linguistics.

Kumar, V., Smith-Renner, A., Findlater, L., Seppi, K., and Boyd-Graber, J. (2019). Why didn't you listen to me? comparing user control of human-in-the-loop topic models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6323–6330, Florence, Italy. Association for Computational Linguistics.

Kwitt, R., Hegenbart, S., and Niethammer, M. (2016). One-shot learning of scene locations via feature trajectory transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 78–86.

Kwon, Y.-S. and Yi, B.-J. (2012). Design and motion planning of a two-module collaborative indoor pipeline inspection robot. *IEEE Transactions on Robotics*, 28(3):681–696.

Lane, D., Brown, K., Petillot, Y., Miguelanez, E., and Patron, P. (2013a). *An Ontology-Based Approach to Fault Tolerant Mission Execution for Autonomous Platforms*, pages 225–255. Springer New York, New York, NY.

Lane, D., Brown, K., Petillot, Y., Miguelanez, E., and Patron, P. (2013b). An ontology-based approach to fault tolerant mission execution for autonomous platforms. In Seto, M., editor, *Marine Robot Autonomy*, pages 225–255. Springer.

Le, H., Sahoo, D., Chen, N. F., and Hoi, S. C. H. (2019). Multimodal transformer networks for end-to-end video-grounded dialogue systems. In *ACL*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, C., Jung, S., Kim, S., and Lee, G. G. (2009). Example-based dialog modeling for practical multi-domain dialog system. *Speech Commun.*, 51(5):466–484.

Lemon, O., Bracy, A., Gruenstein, A., and Peters, S. (2001). The witas multi-modal dialogue system i. In *Proceedings of the Seventh European Conference on Speech Communication and Technology.* 7th European Conference on Speech Communication and Technology, Eurospeech 2001 ; Conference date: 03-09-2001 Through 07-09-2001.

Levin, E., Pieraccini, R., and Eckert, W. (2000). A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In Croft, B. W. and van Rijsbergen, C. J., editors, *SIGIR '94*, pages 3–12, London. Springer London.

Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., and Batra, D. (2017). Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. (2016a). A persona-based neural conversation model. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics, pages 994–1003. Association for Computational Linguistics.

Li, J., Yang, J., Hertzmann, A., Zhang, J., and Xu, T. (2021). Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2388–2399.

Li, L., He, H., and Williams, J. D. (2014). Temporal supervised learning for inferring a dialog policy from example conversations. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pages 312–317.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. (2019). Visualbert: A simple and performant baseline for vision and language. In *Arxiv*.

Li, S., Scalise, R., Admoni, H., Rosenthal, S., and Srinivasa, S. S. (2016b). Spatial references and perspective in natural language instructions for collaborative manipulation. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 44–51.

Li, X., Yin, X., Li, C., Hu, X., Zhang, P., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. (2020). Oscar: Object-semantics aligned pretraining for vision-language tasks. *ECCV 2020*.

Liao, Y., Liu, S., Li, G., Wang, F., Chen, Y., Qian, C., and Li, B. (2020). A real-time cross-modality correlation filtering method for referring expression comprehension. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10877–10886.

Lin, X., Li, G., and Yu, Y. (2021). Scene-intuitive agent for remote embodied visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7036–7045.

Liu, Z., Wang, J., Gong, S., Tao, D., and Lu, H. (2019). Deep reinforcement active learning for human-in-the-loop person re-identification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6121–6130.

Lopatovska, I., Rink, K., Knight, I., Raines, K., Cosenza, K., Williams, H., Sorsche, P., Hirsch, D., Li, Q., and Martinez, A. (2019). Talk to me: Exploring user interactions with the amazon alexa. *Journal of Librarianship and Information Science*, 51(4):984–997.

Lowe, R., Pow, N., Serban, I., Charlin, L., Liu, C.-W., and Pineau, J. (2017). Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue and Discourse*, 8:31–65.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 13–23. Curran Associates, Inc.

Luan, Y., Ji, Y., and Ostendorf, M. (2016). Lstm based conversation models.

Lyons, K., Skeels, C., and Starner, T. (2005). Providing support for mobile calendaring conversations: A wizard of oz evaluation of dual–purpose speech. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices amp; Services*, MobileHCI '05, page 243–246, New York, NY, USA. Association for Computing Machinery.

Ma, C., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., and Xiong, C. (2019). Self-monitoring navigation agent via auxiliary progress estimation. *CoRR*, abs/1901.03035.

MacMahon, M., Stankiewicz, B., and Kuipers, B. (2006). Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1475–1482. AAAI Press.

Mairesse, F., Gašić, M., Jurčíček, F., Keizer, S., Thomson, B., Yu, K., and Young, S. (2010). Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1552–1561, Uppsala, Sweden. Association for Computational Linguistics.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., and Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20.

Marge, M., Bonial, C., Foots, A., Hayes, C., Henry, C., Pollard, K., Artstein, R., Voss, C., and Traum, D. (2017). Exploring variation of natural human commands to a robot in a collaborative navigation task. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 58–66, Vancouver, Canada. Association for Computational Linguistics.

Matuszek, C., Bo, L., Zettlemoyer, L., and Fox, D. (2014). Learning from unscripted deictic gesture and language for human-robot interactions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).

Mehta, N. and Goldwasser, D. (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)*, pages 1962–1967, Minneapolis, Minnesota. Association for Computational Linguistics.

Mei, H., Bansal, M., and Walter, M. R. (2017). Coherent dialogue with attention-based language models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3252–3258. AAAI Press.

Miguelanez, E., Patron, P., Brown, K. E., Petillot, Y. R., and Lane, D. M. (2011). Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. *IEEE Transactions on Knowledge and Data Engineering*, 23(5):759–773.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. H., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, E. G., Matsakis, N. E., and Viola, P. A. (2000). Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 464–471 vol.1.

Min, S. Y., Chaplot, D. S., Ravikumar, P., Bisk, Y., and Salakhutdinov, R. (2022). FILM: Following Instructions in Language with Modular Methods. In *The Tenth International Conference on Learning Representations*.

Misra, D., Bennett, A., Blukis, V., Niklasson, E., Shatkhin, M., and Artzi, Y. (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*, pages 2667–2678, Brussels, Belgium. Association for Computational Linguistics.

Misra, D., Langford, J., and Artzi, Y. (2017). Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015, Copenhagen, Denmark. Association for Computational Linguistics.

Misu, T. (2018). Situated reference resolution using visual saliency and crowdsourcing-based priors for a spoken dialog system within vehicles. *Computer Speech and Language*, 48:1 – 14.

Moratz, R., Fischer, K., and Tenbrink, T. (2001). Cognitive modeling of spatial reference for human-robot interaction. *International Journal on Artificial Intelligence Tools*, 10:589–611.

Nagaraja, V. K., Morariu, V. I., and Davis, L. S. (2016). Modeling context between objects for referring expression understanding. In *ECCV*.

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., and Kawatsuma, S. (2013). Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63.

Novikova, J., Dušek, O., and Rieser, V. (2017). The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrücken, Germany.

O'Grady, W. (2005). *How Children Learn Language*. Cambridge Approaches to Linguistics. Cambridge University Press.

Oviatt, S., Coulston, R., and Lunsford, R. (2004). When do we interact multimodally?: cognitive load and multimodal communication patterns. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 129–136, New York, NY, USA. ACM.

Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramuthu, R., Tür, G., and Hakkani-Tür, D. (2021). Teach: Task-driven embodied agents that chat. *CoRR*, abs/2110.00534.

Pashevich, A., Schmid, C., and Sun, C. (2021). Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15942–15952.

Pateras, C., Dudek, G., and De Mori, R. (1995). Understanding referring expressions in a person-machine spoken dialogue. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 197–200 vol.1.

Paz-Argaman, T. and Tsarfaty, R. (2019). RUN through the streets: A new dataset and baseline models for realistic urban navigation. 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)*, pages 6449–6455, Hong Kong, China. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Persson, M., Duckett, T., Valgren, C., and Lilienthal, A. (2007). Probabilistic semantic mapping with a virtual sensor for building/nature detection. In *2007*

International Symposium on Computational Intelligence in Robotics and Automation, pages 236–242.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Petillot, Y., Sotzing, C., Patron, P., Lane, D., and Cartright, J. (2009). Multiple system collaborative planning and sensing for autonomous platforms with shared and distributed situational awareness. In *Proceedings of the AUVSI's Unmanned Systems Europe, La Spezia, Italy.*

Petscharnig, S., Lux, M., and Chatzichristofis, S. (2017). Dimensionality reduction for image features using deep learning and autoencoders. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pages 23:1–23:6, New York, NY, USA. ACM.

Phillips, E., Ososky, S., Grove, J., and Jentsch, F. (2011). From tools to teammates: Toward the development of appropriate mental models for intelligent robots. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 55(1):1491–1495.

Platonov, G., Schubert, L., Kane, B., and Gindi, A. (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*, pages 128–131, 1st virtual meeting. Association for Computational Linguistics.

Plummer, B. A., Kordas, P., Kiapour, M. H., Zheng, S., Piramuthu, R., and Lazebnik, S. (2018). Conditional image-text embedding networks. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 258–274, Cham. Springer International Publishing.

Purver, M., Eshghi, A., and Hough, J. (2011). Incremental semantic construction in a dialogue system. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011).*

Purver, M., Hough, J., and Gregoromichelaki, E. (2014). *Dialogue and Compound Contributions.*

Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W. Y., Shen, C., and van den Hengel, A. (2020). Reverie: Remote embodied visual referring expression in real indoor environments. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9979–9988.

Qian, K., Chozhiyath Raman, P., Li, Y., and Popa, L. (2020). Partner: Human-in-the-loop entity name understanding with deep learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13634–13635.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63:1872–1897.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Rajman, M., Ailomaa, M., Lisowska, A., Melichar, M., and Armstrong, S. (2006). Extending the Wizard of Oz methodologie for multimodal language-enabled systems. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Ren, S., He, K., Girshick, R., and Sun, J. (2015a). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ren, S., He, K., Girshick, R., and Sun, J. (2015b). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework:

Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Robotics, C. (2011). Husky unmanned ground vehicle. `https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/`. Accessed: 2010-09-30.

Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., and Tomlinson, B. (2010). Who are the crowdworkers?: Shifting demographics in mechanical turk. In *Proceedings of the CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 2863–2872, New York, NY, USA. ACM.

Sadhu, A., Chen, K., and Nevatia, R. (2019). Zero-shot grounding of objects from natural language queries. In *The IEEE International Conference on Computer Vision (ICCV)*.

Salin, E., Farah, B., Ayache, S., and Favre, B. (2022). Are Vision-Language Transformers Learning Multimodal Representations? A Probing Perspective. In *AAAI 2022*, Vancouver, Canada.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., and Batra, D. (2019). Habitat: A platform for embodied AI research. *CoRR*, abs/1904.01201.

Schlangen, D. (2016). Grounding, Justification, Adaptation: Towards Machines That Mean What They Say. In *Proceedings of the 20th Workshop on the Semantics and Pragmatics of Dialogue (JerSem)*.

Serban, I., Lowe, R., Charlin, L., and Pineau, J. (2016a). Generative deep neural networks for dialogue: A short review. *ArXiv*, abs/1611.06216.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016b). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 3776–3783. AAAI Press.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In Hutter, M. and Siegwart, R., editors, *Field and Service Robotics*, pages 621–635, Cham. Springer International Publishing.

Shalyminov, I. (2020). *Data-Efficient Methods for Dialogue Systems*. PhD thesis, Heriot-Watt University, Edinburgh, Scotland UK.

Shang, L., Lu, Z., and Li, H. (2015). Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China. Association for Computational Linguistics.

Shen, H. T., Zhu, X., Zhang, Z., Wang, S.-H., Chen, Y., Xu, X., and Shao, J. (2021). Heterogeneous data fusion for predicting mild cognitive impairment conversion. *Information Fusion*, 66:54–63.

Shridhar, M., Manuelli, L., and Fox, D. (2021a). Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. (2019). ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. (2020). ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. (2021b). Alfworld: Aligning text and embodied environments for interactive learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shukla, A. and Karki, H. (2016). Application of robotics in onshore oil and gas industry—a review part i. volume 75, pages 490–507.

Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.

Singh, S., Litman, D., Kearns, M., and Walker, M. (2002). Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *J. Artif. Int. Res.*, 16(1):105–133.

Skantze, G. and Hjalmarsson, A. (2010). Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan. Association for Computational Linguistics.

Skočaj, D., Vrečko, A., Mahnič, M., Janíček, M., Kruijff, G.-J. M., Hanheide, M., Hawes, N., Wyatt, J. L., Keller, T., Zhou, K., Zillich, M., and Kristan, M. (2016). An integrated system for interactive continuous learning of categorical knowledge. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(5):823–848.

Smith, A., Kumar, V., Boyd-Graber, J., Seppi, K., and Findlater, L. (2018). Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, page 293–304, New York, NY, USA. Association for Computing Machinery.

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017). Semantic scene completion from a single depth image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–198, Los Alamitos, CA, USA. IEEE Computer Society.

Strauss, A. L. (1987). *Qualitative analysis for social scientists.* Cambridge University Press.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2020). Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.

Suhr, A., Yan, C., Schluger, J., Yu, S., Khader, H., Mouallem, M., Zhang, I., and Artzi, Y. (2019a). 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)*, pages 2119–2130, Hong Kong, China. Association for Computational Linguistics.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. (2019b). A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., and Maglogiannis, I., editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham. Springer International Publishing.

Tan, H. and Bansal, M. (2018). Source-target inference models for spatial instruction understanding. In *AAAI*.

Tan, H. and Bansal, M. (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)*, pages 5099–5110, Hong Kong, China. Association for Computational Linguistics.

Tan, H., Yu, L., and Bansal, M. (2019). Learning to navigate unseen environments: Back translation with environmental dropout. 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)*, pages 2610–2621, Minneapolis, Minnesota. Association for Computational Linguistics.

Technology, T. I. o. E. a. (2015). Engineering and technology skills and demand in industry 2015 survey. overview of issues and trends from 2015 survey. Technical report. Accessed on 13th March 2018.

Tellex, S., Gopalan, N., Kress-Gazit, H., and Matuszek, C. (2020). Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):25–55.

Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S., and Roy, N. (2011a). Approaching the symbol grounding problem with probabilistic graphical models.

Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S., and Roy, N. (2011b). Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 1507–1514. AAAI Press.

Thomason, J., Murray, M., Cakmak, M., and Zettlemoyer, L. (2019a). Vision-and-dialog navigation. *CoRR*, abs/1907.04957.

Thomason, J., Padmakumar, A., Sinapov, J., Walker, N., Jiang, Y., Yedidsion, H., Hart, J. W., Stone, P., and Mooney, R. J. (2019b). Improving grounded natural language understanding through human-robot dialog. In *ICRA*, pages 6934–6941.

Thomason, J., Sinapov, J., Svetlik, M., Stone, P., and Mooney, R. J. (2016). Learning multi-modal grounded linguistic semantics by playing "i spy". In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 3477–3483. AAAI Press.

Thompson, H. S., Anderson, A., Bard, E. G., Doherty-Sneddon, G., Newlands, A., and Sotillo, C. (1993). The hcrc map task corpus: Natural dialogue for speech recognition. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, page 25–30, USA. Association for Computational Linguistics.

Torruella, J. and Capsada, R. (2013). Lexical statistics and tipological structures: A measure of lexical richness. *Procedia - Social and Behavioral Sciences*, 95:447 – 454. Corpus Resources for Descriptive and Applied Studies. Current Challenges and Future Directions: Selected Papers from the 5th International Conference on Corpus Linguistics (CILC2013).

ur Rehman, S., Tu, S., Waqas, M., Huang, Y., ur Rehman, O., Ahmad, B., and

Ahmad, S. (2019). Unsupervised pre-trained filter learning approach for efficient convolution neural network. *Neurocomputing*, 365:171–190.

Van Schijndel, M., Mueller, A., and Linzen, T. (2019). Quantity doesn't buy quality syntax with neural language models. 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)*, pages 5831–5837, Hong Kong, China. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017a). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., and Polosukhin, I. (2017b). Attention is all you need. In *NeurIPS*.

Vigliocco, G., Perniss, P., and Vinson, D. (2014). Language as a multimodal phenomenon: Implications for language learning, processing and evolution. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 369.

Vries, H. d., Shuster, K., Batra, D., Parikh, D., Weston, J., and Kiela, D. (2018). Talk the walk: Navigating new york city through grounded dialogue. *CoRR*, abs/1807.03367.

Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. (1997). Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, ACL '98, pages 271–280, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wallace, E., Rodriguez, P., Feng, S., Yamada, I., and Boyd-Graber, J. (2019). Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Wang, L., Li, Y., and Lazebnik, S. (2017). Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP.

Wang, Z. J., Choi, D., Xu, S., and Yang, D. (2021). Putting humans in the natural language processing loop: A survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 47–52, Online. Association for Computational Linguistics.

Wen, T.-H., Vandyke, D., Mrkšić, N., Gasic, M., Rojas Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 438–449. Association for Computational Linguistics.

Weng, F., Varges, S., Raghunathan, B., Ratiu, F., Pon-Barry, H., Lathrop, B., Zhang, Q., Bratt, H., Scheideck, T., Xu, K., Purver, M., Mishra, R., Lien, A., Raya, M., Peters, S., Meng, Y., Russell, J., Cavedon, L., Shriberg, E., and Prieto, R. (2006). Chat: a conversational helper for automotive tasks.

Werner, S., Krieg-Brückner, B., Mallot, H. A., Schweizer, K., and Freksa, C. (1997). Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation. In *GI Jahrestagung*.

Williams, J. (2008). The best of both worlds: Unifying conventional dialog systems and pomdps. pages 1173–1176.

Williams, J., Raux, A., and Henderson, M. (2016). The dialog state tracking challenge series: A review. *Dialogue Discourse*.

Williams, J. D., Asadi, K., and Zweig, G. (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*, pages 665–677, Vancouver, Canada. Association for Computational Linguistics.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Wolf, T., Sanh, V., Chaumond, J., and Delangue, C. (2019). Transfertransfo: A transfer learning approach for neural network based conversational agents.

Wolfe, J. (1994). Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin Review*, 1:202–238.

Wong, C., Yang, E., Yan, X.-T., and Gu, D. (2017). An overview of robotics and autonomous systems for harsh environments. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6.

Wu, X., Xiao, L., Yixuan, S., Zhang, J., Ma, T., and He, L. (2021). A survey of human-in-the-loop for machine learning.

Wu, Y., Wu, Y., Gkioxari, G., and Tian, Y. (2018). Building generalizable agents with a realistic and rich 3d environment. In *Proceedings of the 6th International-Conference on Learning Representations*, ICLR '18.

Xiang, J., Wang, X., and Wang, W. Y. (2020). Learning to stop: A simple yet effective approach to urban vision-language navigation. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 699–707. Association for Computational Linguistics.

Xinpeng, C., Ma, L., Chen, J., Jie, Z., Liu, W., and Luo, J. (2018). Real-time referring expression comprehension by single-stage grounding network.

Xiong, W., Guo, X., Yu, M., Chang, S., Zhou, B., and Wang, W. Y. (2018). Scheduled policy optimization for natural language communication with intelligent agents. In *Proceedings of IJCAI*, pages 4503–4509.

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015a). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2048–2057. JMLR.org.

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015b). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2048–2057. JMLR.org.

Yan, C., Misra, D., Bennnett, A., Walsman, A., Bisk, Y., and Artzi, Y. (2018). Chalet: Cornell house agent learning environment.

Yang, F., Yang, H., Fu, J., Lu, H., and Guo, B. (2020). Learning texture transformer network for image super-resolution. In *Proceedings of the conference on Computer Vision and Pattern Recognition (CVPR).*

Yang, W., Wang, X., Farhadi, A., Gupta, A., and Mottaghi, R. (2019a). Visual semantic navigation using scene priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net.

Yang, Y., Li, Y., and Quan, X. (2021). Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14230–14238.

Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., and Luo, J. (2019b). A fast and accurate one-stage approach to visual grounding. pages 4682–4692.

Yin, Z., Chang, K.-h., and Zhang, R. (2017). Deepprobe: Information directed sequence understanding and chatbot design via recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 2131–2139, New York, NY, USA. Association for Computing Machinery.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop.

Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., and Berg, T. L. (2018). Mattnet: Modular attention network for referring expression comprehension. In *CVPR.*

Yu, Y., Eshghi, A., and Lemon, O. (2016). 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*, pages 339–349, Los Angeles. Association for Computational Linguistics.

Yu, Y., Eshghi, A., and Lemon, O. (2017). Learning how to learn: An adaptive dialogue agent for incrementally learning visually grounded word meanings. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 10–19, Vancouver, Canada. Association for Computational Linguistics.

Zaib, M., Sheng, Q. Z., and Emma Zhang, W. (2020). A short survey of pre-trained language models for conversational ai-a new age in nlp. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '20, New York, NY, USA. Association for Computing Machinery.

Zeng, Y., Fu, J., and Chao, H. (2020). Learning joint spatial-temporal transformations for video inpainting. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhang, H., Niu, Y., and Chang, S.-F. (2018a). Grounding referring expressions in images by variational context. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, H., Niu, Y., and Chang, S.-F. (2018b). Grounding referring expressions in images by variational context. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4158–4166.

Zhang, L., Wang, L., Zhang, X., Shen, P., Bennamoun, M., Zhu, G., Shah, S. A. A., and Song, J. (2018c). Semantic scene completion with dense crf from a single depth image. *Neurocomputing*, 318:182–195.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press.

Zhao, S., Liu, Z., Lin, J., Zhu, J.-Y., and Han, S. (2020). Differentiable augmentation for data-efficient gan training. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

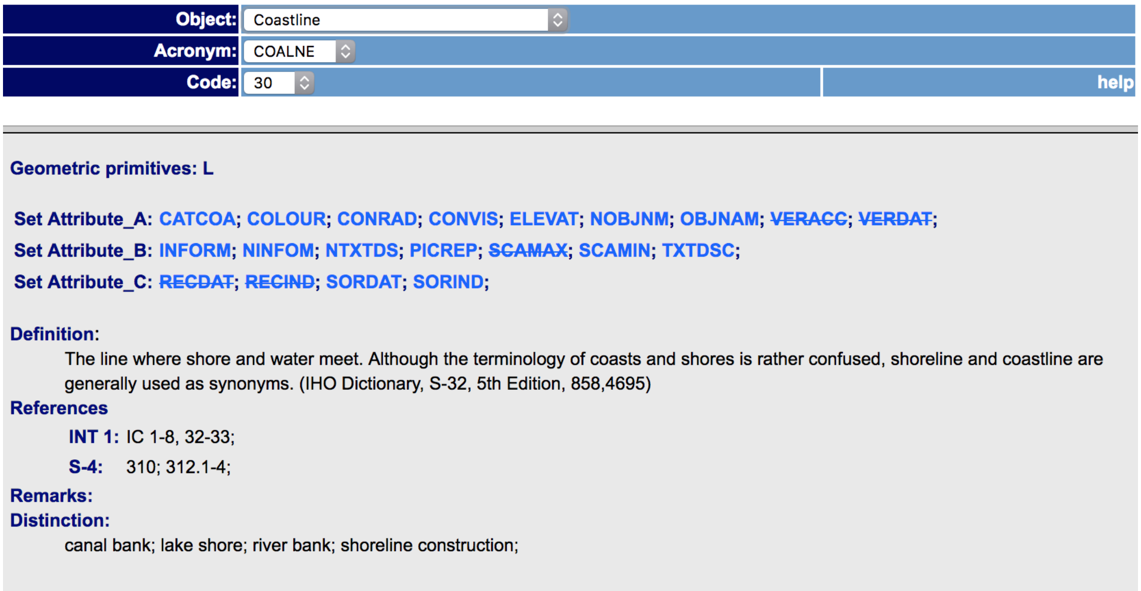
Zhuang, B., Wu, Q., Shen, C., Reid, I., and Hengel, A. (2018). Parallel attention: a unified framework for visual object discovery through dialogs and queries. In Forsyth, D., Laptev, I., Oliva, A., and Ramanan, D., editors, *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 4252–4261, United States of America. IEEE, Institute of Electrical and Electronics Engineers.

Appendix A

S-57 Data Format and Generation of Data

A.1 The S-57 Data format

Examples of the S-57 architecture, used in the ENC Charts presented in this thesis are shown in Figure A.1 and A.2.



Object: Coastline

Acronym: COALNE

Code: 30

Geometric primitives: L

Set Attribute_A: CATCOA; COLOUR; CONRAD; CONVIS; ELEVAT; NOBJNM; OBJNAM; ~~VERACC~~; ~~VERDAT~~;

Set Attribute_B: INFORM; NINFOM; NTXTDS; PICREP; ~~SCAMAX~~; SCAMIN; TXTDSC;

Set Attribute_C: ~~RECDAT~~; ~~RECIND~~; SORDAT; SORIND;

Definition:

The line where shore and water meet. Although the terminology of coasts and shores is rather confused, shoreline and coastline are generally used as synonyms. (IHO Dictionary, S-32, 5th Edition, 858,4695)

References

INT 1: IC 1-8, 32-33;

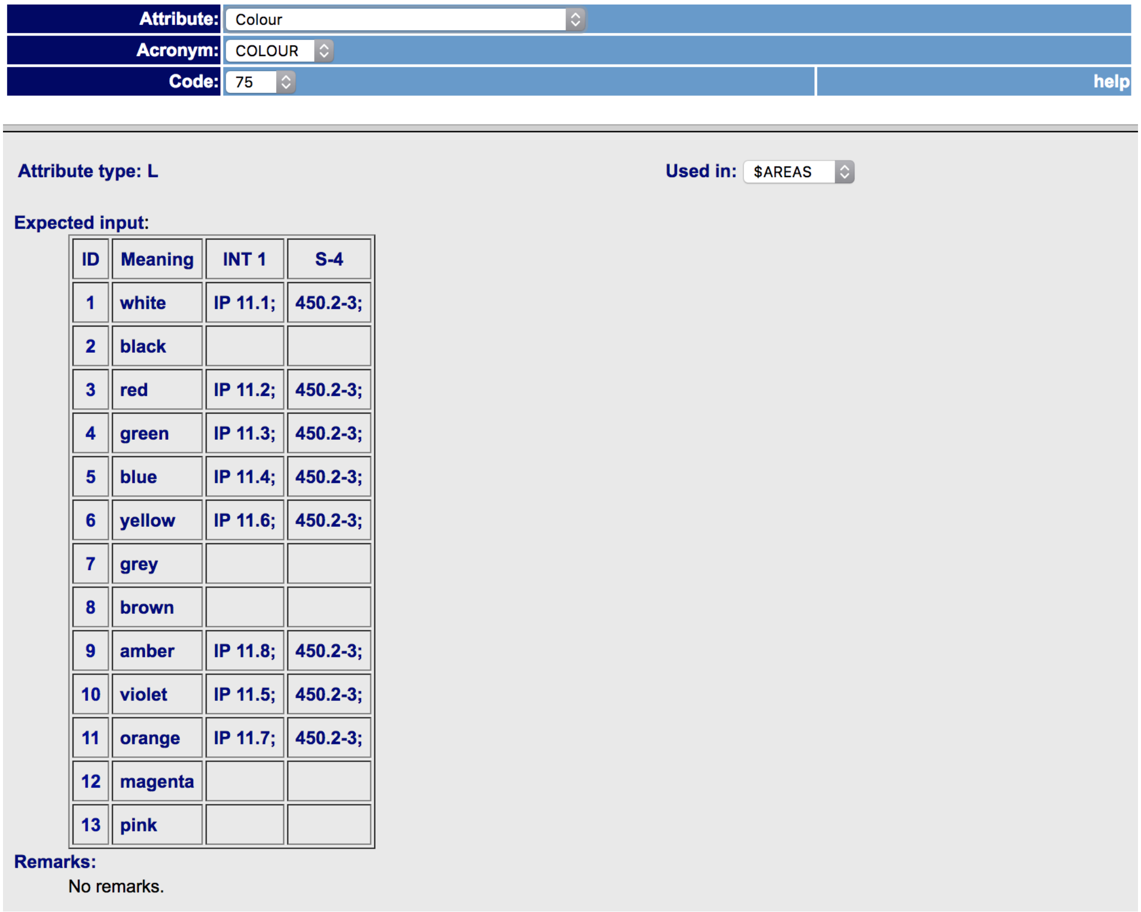
S-4: 310; 312.1-4;

Remarks:

Distinction:

canal bank; lake shore; river bank; shoreline construction;

Figure A.1: Example of S-57 object "Coastline" taken from `http://www.s-57.com/`.



Attribute: Colour

Acronym: COLOUR

Code: 75

Attribute type: L

Used in: $AREAS

Expected input:

ID	Meaning	INT 1	S-4
1	white	IP 11.1;	450.2-3;
2	black		
3	red	IP 11.2;	450.2-3;
4	green	IP 11.3;	450.2-3;
5	blue	IP 11.4;	450.2-3;
6	yellow	IP 11.6;	450.2-3;
7	grey		
8	brown		
9	amber	IP 11.8;	450.2-3;
10	violet	IP 11.5;	450.2-3;
11	orange	IP 11.7;	450.2-3;
12	magenta		
13	pink		

Remarks:

No remarks.

Figure A.2: Example of S-57 attribute "Colour" taken from `http://www.s-57.com/`.

A.2 Generating RENCI from the WOz study

We fill the templated sentence randomly with the right type of word that goes in. The types of words are:

1. mbjname1: Movable object, e.g. drone
2. dist1: Distance, e.g. 200
3. brng1: Bearing, e.g. southwest
4. obj1: Object/landmark, e.g. Starbucks/ coastline

We provide the necessary annotation depending on the template. We split the template, words and annotations with the "@@" symbol.

For instance, in the template "*give me a* {} {} {} *of* {} @@*mbjname1-dist1-brng1-obj1*@@*obj1-dist1-brng1*", the colours represent, (1) template, (2) words and (3) annotations respectively.

Below we show the list of templates we collected from the WoZ study and Seebyte expert separately:

```
give me a {} {} {} of {} @@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

and another {} {} {} of {}@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

{} {} {} {} of {}@@verb-mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

Move the {} {} {} of the {}@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

{} {} {} {} @@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

{} {} at {} @@verb-mbjname1-obj1@@obj1

Create‘‘{}’’ {} {} of {}, 200m wide, 500m high
@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

new robot called‘‘{}’’ starts {} {} from‘‘{}’’
point1@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

please {} a {} near {}@@verb-mbjname1-obj1@@obj1

add a {} near the {}@@mbjname1-obj1@@obj1

please add a {} {} {} of {}@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

Create a second {} {} {} of the
{}.@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

move both {}s {} {} of the {}@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1
```

```
plan a {} {} {} of the {}@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

{} {} {} {} from the {}@@verb-mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

Create an {} along the {} and move it {} by
{}@@mbjname1-obj1-brng1-dist1@@obj1-dist1-brng1

create two {}s {} {} of the {}@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

make a {} {} {} from the {}.@@mbjname1-dist1-brng1-obj1@@obj1-dist1-brng1

{} an {} around {}@@verb-mbjname1-obj1@@obj1

add a launch and recovery at {}@@obj1@@obj1

{} has a convenient infrastructure to launch and recover the

vehicle.@@obj1@@obj1

Keep off {} and danger.@@obj1@@obj1

Create a {} on {}@@mbjname1-obj1@@obj1

Go to {}@@obj1@@obj1

Send {} to the centre of {}@@mbjname1-obj1@@obj1

I want one {} to patrol {}@@mbjname1-obj1@@obj1

Show me {}@@obj1@@obj1
```

```
Patrol along the {}@@obj1@@obj1

Go {} of floating {}.@@brng1-obj1@@obj1-brng1

Go and Inspect the {}.@@obj1@@obj1

Ensure the {} stays in {}.@@mbjname1-obj1@@obj1

Do not go over the {}@@obj1@@obj1
```

Below the words/values for filling the templated sentences are being shown.

```
meteoBrngWords = {

   ‘0’:[‘north’,‘up’,’above’],

    ‘45’:[‘ne’,‘north east’,‘northeast’,‘north-east’],

    ‘90’:[‘east’,‘right’],

    ‘135’:[‘se’,‘south east’,‘southeast’,‘south-east’],

    ‘180’:[‘south’,‘down’,‘below’],

    ‘225’:[‘sw’,‘south west’,‘southwest’,‘south-west’],

    ‘270’:[‘west’,‘left’],

    ‘315’:[‘nw’,‘north west’,‘northwest’,‘north-west’],
    }

dist_type = [‘m’,‘ m’,‘ meters’]

OBJECTS = [

   ‘‘none’’,

   ‘‘survey area’’,

   ‘‘exclusion zone’’,

   ‘‘surface vehicle’’,

   ‘‘aerial vehicle’’,

   ‘‘underwater vehicle’’

]

objects_to_draw = {

   ‘‘survey area’’ :‘‘survey’’,

   ‘‘exclusion zone’’:‘‘exclusion’’,

   ‘‘surface vehicle’’:‘‘surface vehicle’’,

   ‘‘aerial vehicle’’:‘‘drone’’,

   ‘‘underwater vehicle’’:‘‘auv’’
```

```
}

robots = [‘underwater vehicle’,‘surface vehicle’,‘drone’, ‘aerial vehicle’ ]

words = {‘verb’ : [‘move’,‘add’, ‘set’, ‘put’, ‘create’, ‘send’]}

list_of_ints = [str(random.randint(0, 1500)) for x in range(0,k)]
```

Appendix B

Human Evaluation

B.1 Experimental Protocol

B.1.1 The Task

You are in charge of some robots to rescue vessels in distress. There are two possibilities:

- You received a distress signal so you can localise the vessel in distress on the chart.
- You received a distress signal but you need to search an area to locate the vessel in distress.

You are controlling three robots:

- Two autonomous ships that can go to the given location to rescue the vessel.
- One drone that can search a large area of the sea to locate the vessel in distress.

You have an Intelligent chatbot, “Hermes” as an assistant to help you tasking the robots. The chatbot understands the English language well and you can text single sentence commands to send the robots to either search for the vessels or to rescue its crew. You are provided with two versions of the nautical charts to direct the robots to the location of interest. The first image shows the mission that needs to be performed and the second is just to help you identify all the landmarks. You may refer to any available landmarks on the chart. The chatbot has the same second chart. Try to use direction and distance with the help of the compass and the scale. If one scenario needs more than one task to be done, feel free to write more than one command. Note, there are some areas that are dangerous and could lead to the loss of the robots. These are blocked out areas in red in the first chart. Examples commands:

- Move Boat 1 30m south from the Main Tower, to board the crew.
- Send Boat 2 450m west north of the Harbour and Search for the vessel.

You are provided with 5 rescue scenarios, where you receive one or more distress signals in each scenario. For the first scenario, there are example commands included to help you familiarise yourself with the task. A key is given for symbols on the first chart.

Do not forget to press submit at the end of every form! Otherwise, all your progress will be lost.

B.1.2 EXAMPLE

🎙	Location of a vessel in distress
🟥	Area to be avoided
🟧	Area to be searched for the vessel in distress
🚢	Autonomous Surface Vehicle
✈	Autonomous Aerial Vehicle

Figure B.1: Compass on above and other symbols used during the experiment below.

Figure B.1 shows a compass and all possible symbols that can be seen in the first image that you receive and describe the mission. Example of acceptable answers:

- Define search area to cover harbour channel.
- Send aerial vehicle to search for vessel in distress in the defined search area.
- Send surface vehicle to help vessel in distress 200m southwest of Brooks 1.
- Send surface vehicle to help vessel in distress 300m east of Brooks 1.

For exclusion zones you can simply type “Exclude the . . . .” or “Add an exclusion zone at. . . ”

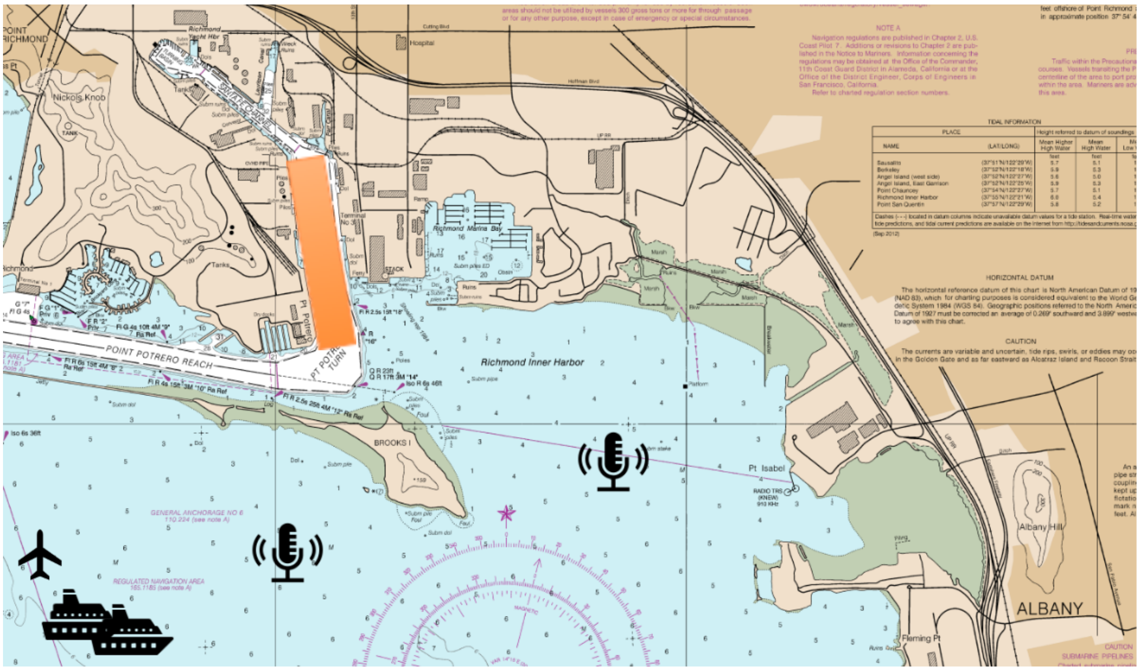


Figure B.2: Example scenario with items. Hermes has the same map but without the mission items on top (Scale: 200m ——).

B.1.3 EXPERIMENT SET-UP

1. **Purpose: This is a formal statement, which encompasses your hypothesis. It is a statement of what question you are trying to answer and what hypothesis you wish to test.**

   Coming from the field of Continual (Iyer et al., 2017) and Human In The Loop learning (Karmakharm et al., 2019), the main Research question we are trying to answer is whether our chatbot “Hermes” can get better online by incorporating human interaction in the learning process in order to improve the performance, “the success rate” of our algorithms’ predictions by complementing the training with new annotated data. Each tester will get every day one of the 5 selected scenarios to interact with. The testers will interact with Hermes with single instructions until the planning for each scenario is complete. If they successfully plan the mission the interaction is marked as successful and they can move on to the next scenario in their next session. If they do not, the interaction is marked unsuccessful and in the next session they will move on to the next scenario but also have access to the previous failed one. This will increase the time they spent per session everyday but they will not have to go through all of the failed scenarios daily as long as they manage to successfully go through all of them at least once. The experiment will run for 14 days so if they work only on one scenario daily, they have an open window to fail at least two times to plan a scenario. However, failing to successfully plan all scenarios does not mean failure of the experiment.

2. **Materials: List all major items needed to carry out your experiment. This list need not be lengthy, but it should include the essentials.** Each subject is required to have their own laptop and internet connection to access our chatbot.
3. **Methods: How will you set up your experiment? What task will the subjects do? How many experimental groups will you have? How will you measure the effect you wish to study? What are your criteria for evaluation? How long will the experiment last? What, if any, questions will you ask the user pre-test and post-test. This testing protocol should be detailed and handed out to each member of the group for consistency.** The experiment will be held completely online using office forms and a chatbot which will run in university servers. Each subject will have to login daily for a week and interact with our chatbot "Hermes" for 15-30 minutes. After each interaction, everyday they will submit a form in which they evaluate the interaction as successful or not. This is the main evaluation for our system to see whether the model is becoming better day by day. Every time an interaction is evaluated as successful the users will gain access to the next scenario until the end of the experiment. If an interaction of a scenario is marked as unsuccessful each user will gain access to

the next scenario but they will have to test again the previous failed one. The testers will have to go through all the scenarios but not necessarily succeed in all of them. Besides that we have a static dataset of two unseen scenarios with a mix of synthetic and real examples that we will evaluate our model daily offline as well. These two will be our main evaluation criteria of our chatbot. The research questions will be answered based mainly on these two metrics. The pre and post test questions will be mainly on their experience with ENC charts and general comments on the interaction with the chatbot that will help us to generate statistics and and error analysis.

4. **Controls: Identify the relevant control(s) condition. Think about the variable(s) you and your group are manipulating. Think about any confounding variables.** There will be only one group of testers. The conditions will be the same across all of them and every day they will have access to the same retrained chatbot.
5. **Data Interpretation: What will be done with the data once it is collected? Data must be organised and summarised so that the scientist himself, and other researchers can determine if the hypothesis has been supported or negated. Results are usually shown in tables and graphs (figures). Statistical analyses are often made to compare experimented and controlled populations.** The data that will be collected everyday and will be used to retrain our chatbot overnight, automatically, without any intervention of the scientists. We will keep track of the testers' ID and save their data in separate folders for future analysis.

B.2 Scenarios

B.2.1 Scenario 1

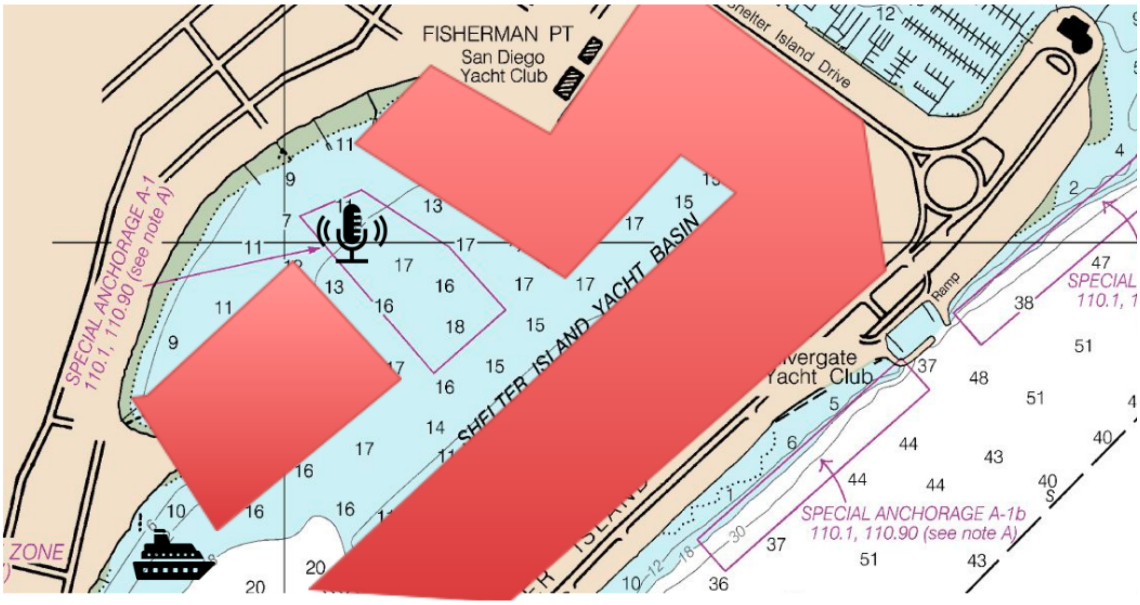


Figure B.3: Scenario 1 as shown to the subjects.

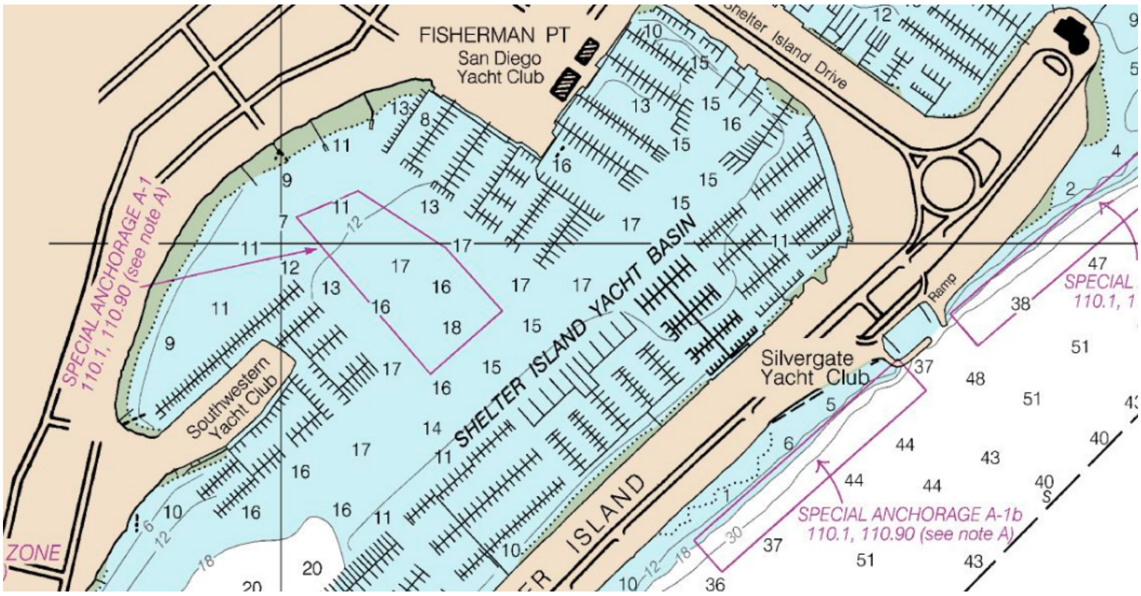


Figure B.4: Scenario 1 as shown to Hermes.

B.2.2 Scenario 3

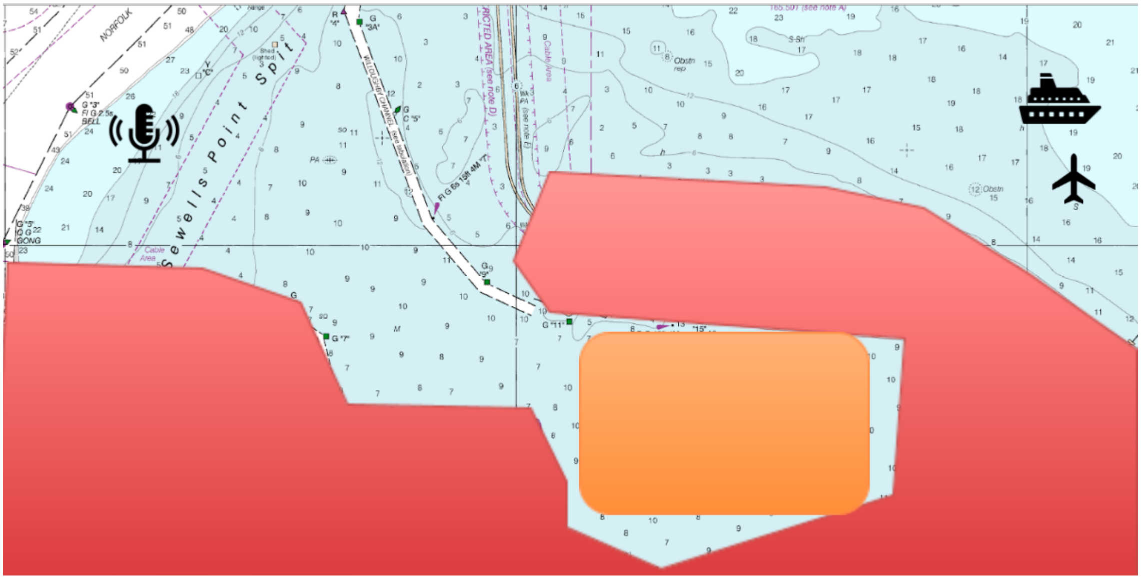


Figure B.5: Scenario 3 as shown to the subjects.

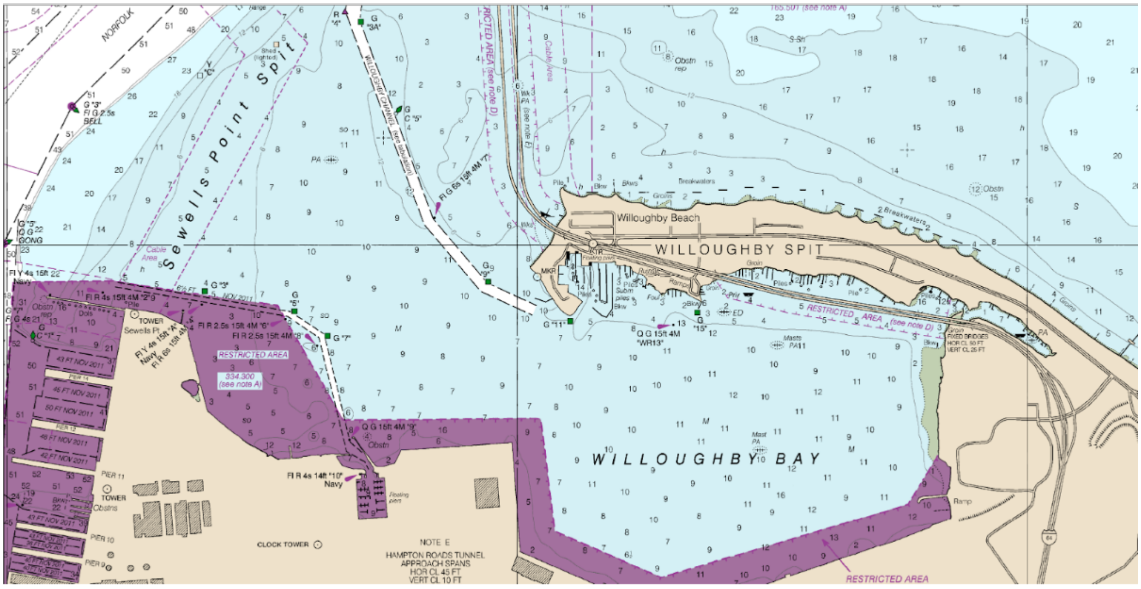


Figure B.6: Scenario 3 as shown to Hermes.

B.2.3 Scenario 4

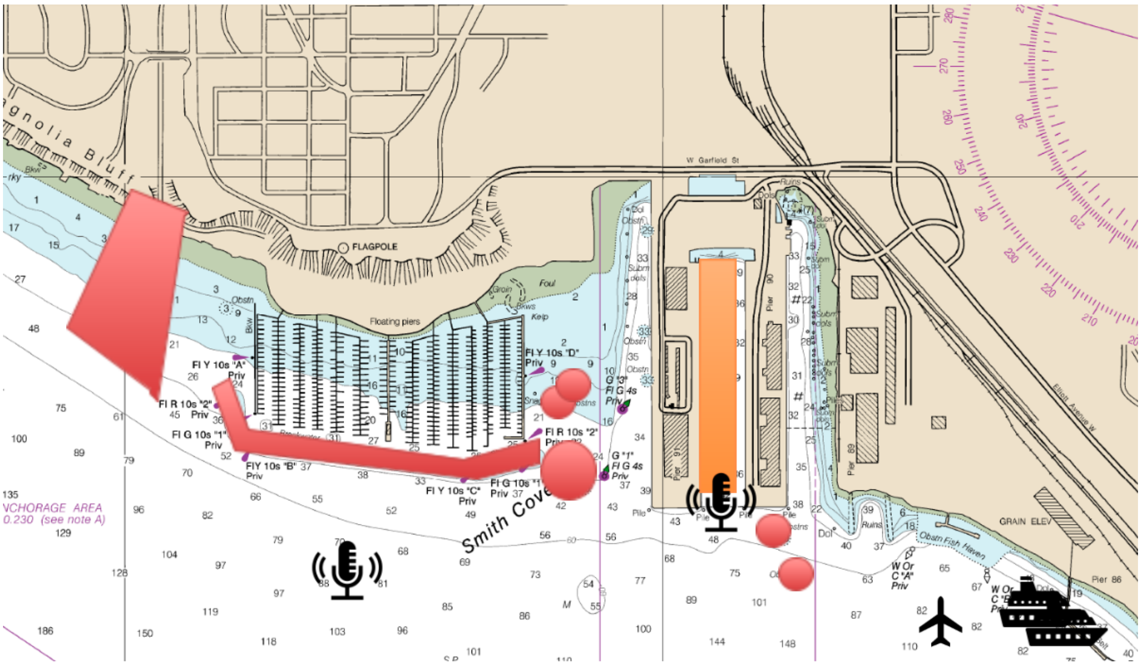


Figure B.7: Scenario 4 as shown to the subjects.

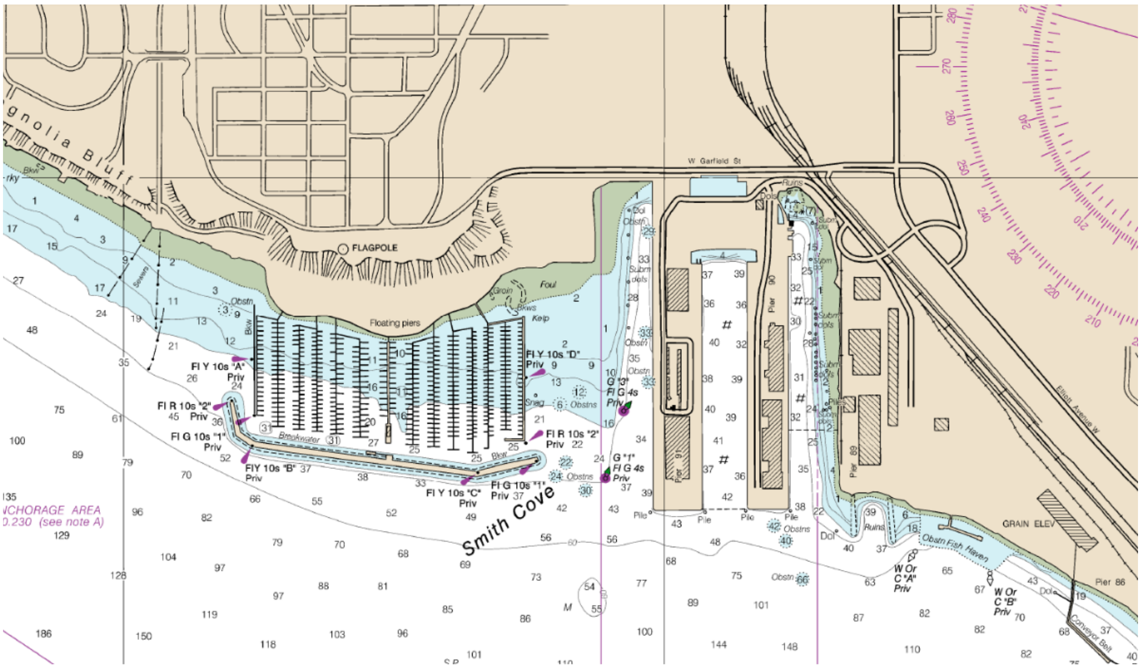


Figure B.8: Scenario 4 as shown to Hermes.

B.2.4 Scenario 6

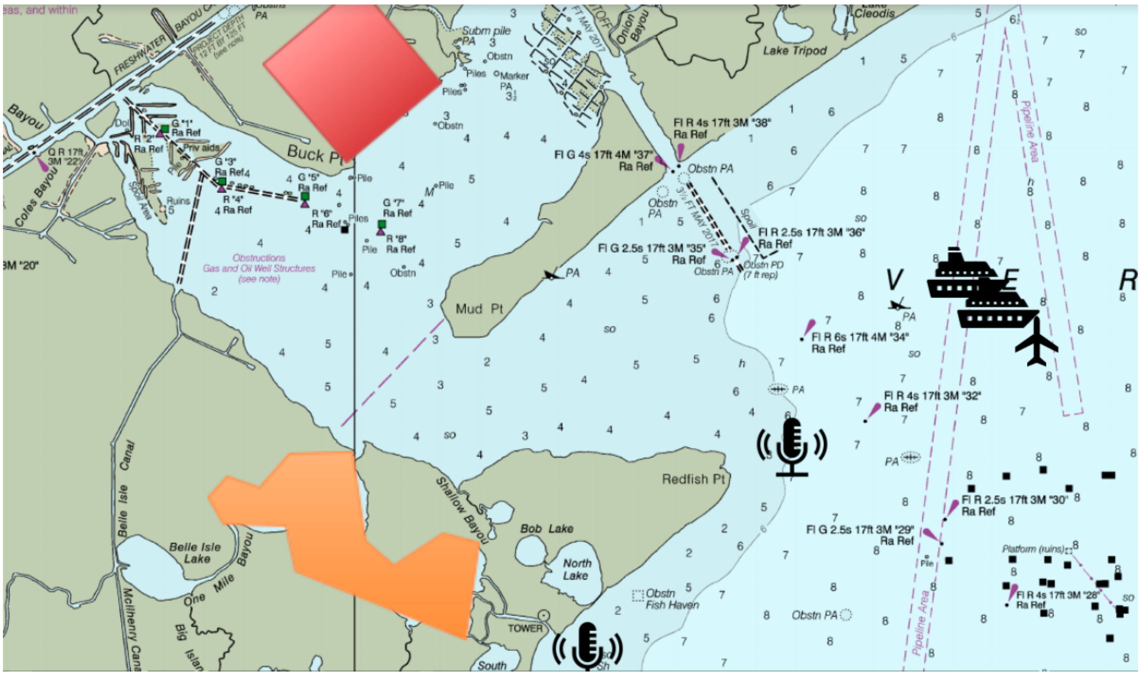


Figure B.9: Scenario 6 as shown to the subjects.

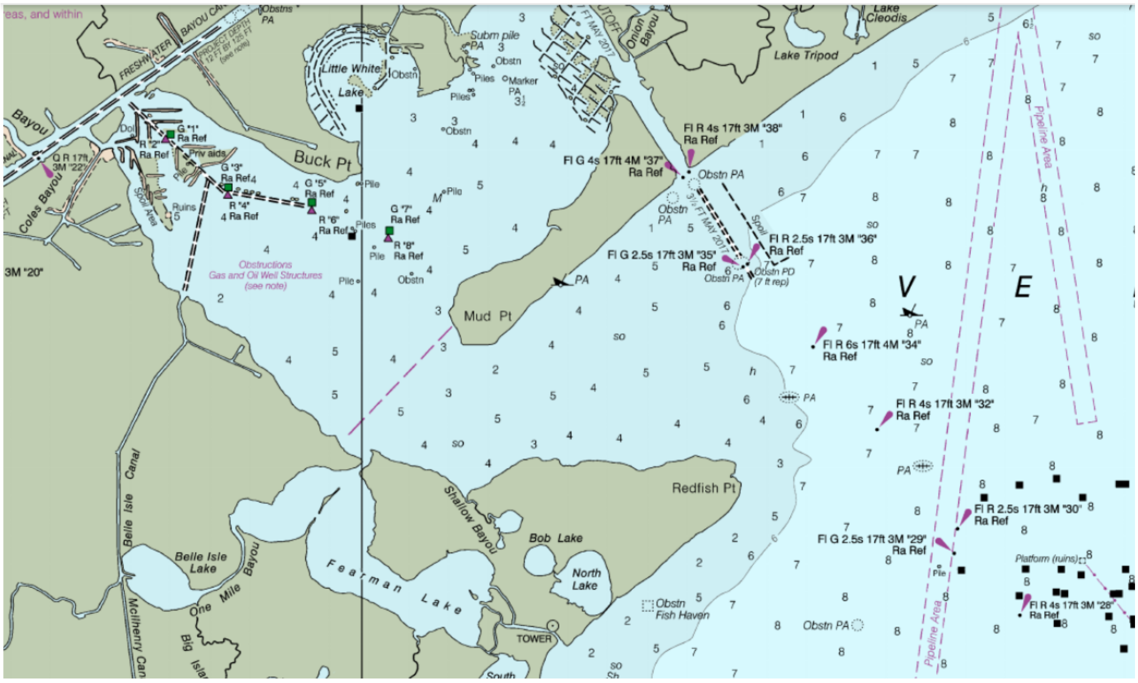


Figure B.10: Scenario 6 as shown to Hermes.

B.2.5 Scenario 7

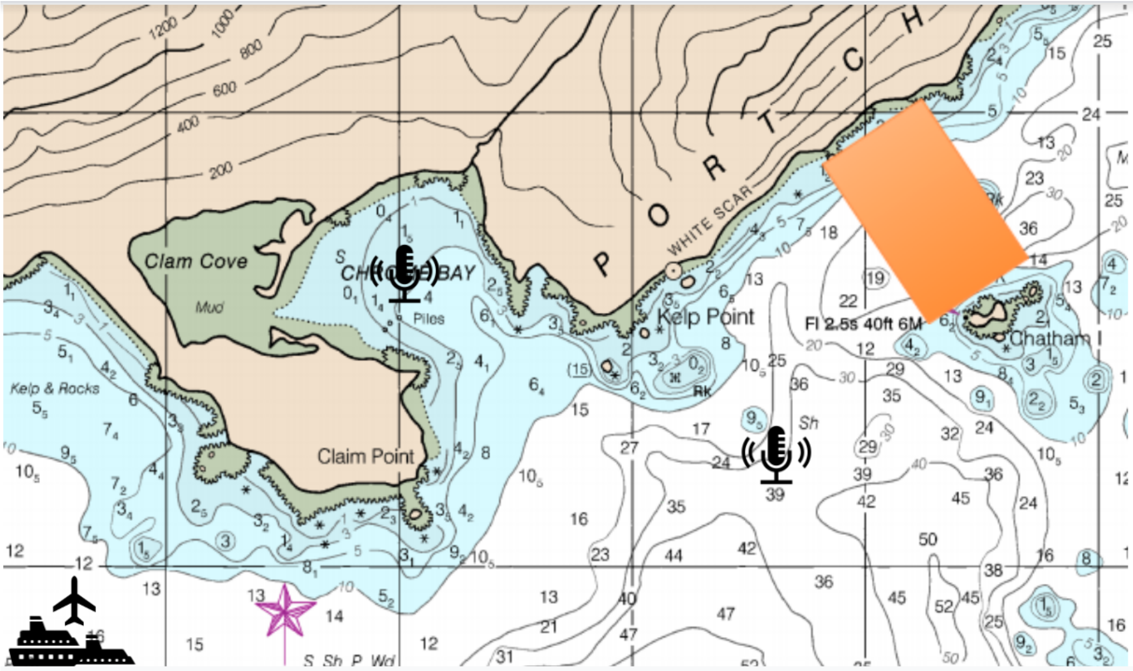


Figure B.11: Scenario 7 as shown to the subjects.

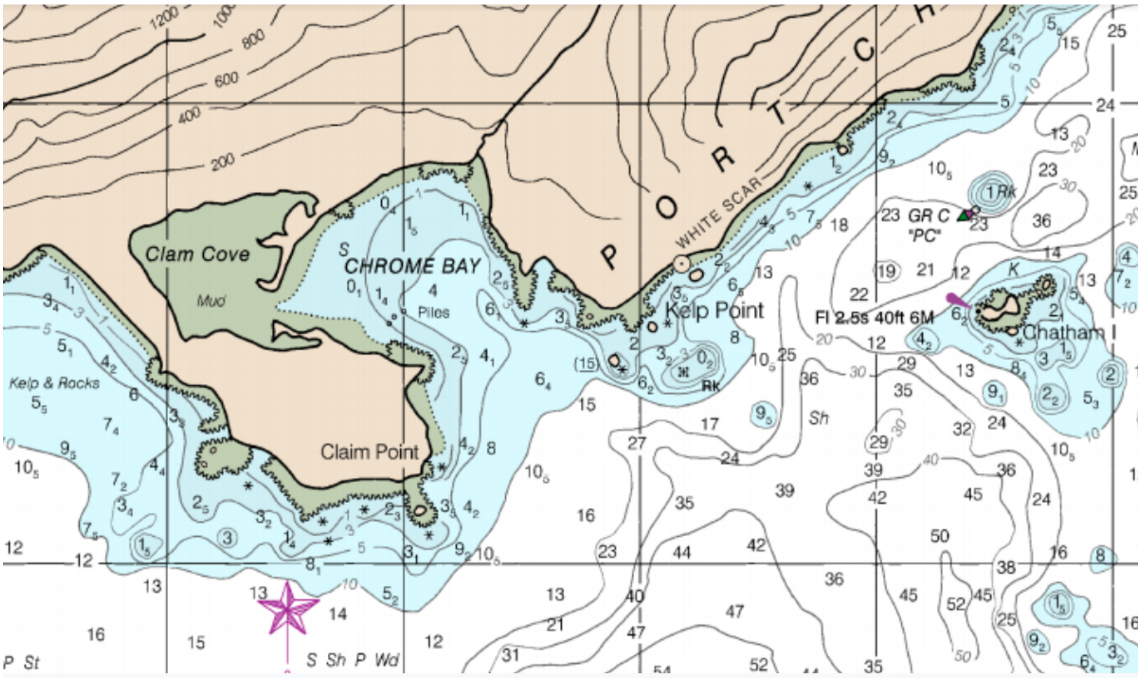


Figure B.12: Scenario 7 as shown to Hermes.

B.3 RASA files

domain.yml

```
version: ‘‘2.0’’
intents:
  - greet
  - goodbye
  - affirm
  - correct
  - deny
  - incorrect
  - show_image
  - out_of_scope
  - call_mapert
  - thank_you
  - your_welcome
  - wait
  - feedback
  - init_mapert
  - ask_annotation
  - get_user_id
  - ask_scenario
  - start_mapert

actions:
  - call_mapert
  - start_mapert
  - get_user_id
  - show_image
  - correct
  - incorrect
  - ask_scenario
  - ask_annotation
entities:
  - annotations
  - start_dist
  - end_dist
  - bearing
  - object
  - start_land
  - end_land
  - landmark

slots:
  annotations:
    type: list
  start_dist:
    type: float
  end_dist:
    type: float
  bearing:
    type: float
  object:
    type: float
  start_land:
    type: float
  end_land:
    type: float
  landmark:
    type: float

responses:
  utter_greet:
  - text: ``Hi I am Hermes! What is the emergency?''
```

```
utter_cheer_up:

- text: ‘‘Here is something to cheer you up:’’

  image: "https://i.imgur.com/nGF1K8f.jpg’’


utter_did_that_help:

- text: ‘‘Did that help you?’’


utter_happy:

- text: ‘‘Great, carry on!’’


utter_goodbye:

- text:

  ‘‘Goodbye! Please follow the link https://forms.office.com/r/90JYFEQLxY.’’



utter_out_of_scope:

- text: Sorry, I can’t handle that request.


utter_your_welcome:

- text: You are welcome!


utter_thank_you:

- text: Thank you, have a wonderful day!


utter_wait:

- text: Processing, please wait...


utter_enqmore:

- text: Can I help you with something else?


utter_feedback:
  - text: Was the interaction successful?


session_config:
  session_expiration_time: 60
  carry_over_slots_to_new_session: true
```

domain.yml

```
version: ‘‘2.0’’
nlu:
- intent: greet
  examples: |
    - hey
    - hello
    - hi
    - hello there
    - good morning
    - good evening
    - moin
    - hey there
    - let’s go
    - hey dude
    - goodmorning
    - goodevening
    - good afternoon

- intent: goodbye
  examples: |
    - good afternoon
    - cu
    - good by
    - see you later
    - good night
    - bye
    - goodbye
    - have a nice day
    - see you around
    - bye bye
    - see you later
    - thats it thank you
    - mission finished
    - i want to give feedback
    - this is over
    - i am done
    - mission complete

- intent: affirm
  examples: |
    - yes
    - y
    - indeed
    - of course
    - that sounds good
    - correct

- intent: deny
  examples: |
    - no
    - n
    - never
    - I don’t think so
    - don’t like that
    - no way
    - not really
- intent: call_mapert
  examples: |
    - Exclude the kelp area
    - Exclude the obstruction.
    - Patrol along the piles.
    - Go East of floating piers.
    - Survey the ruins
    - Exclude the sewer area
    - Inspect between the Smith Cove Buoy 1 and 3, and the coastline
    - send a robot 400m south of the fisherman pt
    - send an auv south 100m of fisherman point
    - Can you send a boat to the anchorage A1?
    - Can you exclude the coast?
    - move drone22 100m right
    - add exclusion zone
    - patrol the area with a drone
    - block the pontoons
    - move boat 300m southwest
    - move surface vehicle southwest
    - move all vehicles
    - add a boat at the claim KeyboardInte
    - move auv22 north of the coast
    - move the survey area 100m southwest
    - boat at survey49
    - survey at bottom
    - go robot there
    - dont go to the pontoons
    - avoid the pontoons
    - add a search area at the willoughby bay
    - search area at the centre



- intent: show_image
  examples: |
    - Show image
    - Show the chart please
    - image
    - chart
    - show the plan
    - show enc chart
    - show

- intent: init_mapert
  examples: |
    - start interaction
    - start
```

rules.yml

```
version: ‘‘2.0’’
rules:

- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye



- rule: add training data when interaction correct
  steps:
  - intent: correct
  - action: correct
- rule: execute annotation tool when incorrect
  steps:
  - intent: ask_annotation
  - action: ask_annotation

- rule: get annotations when incorrect
  steps:
  - intent: incorrect
  - action: incorrect

- rule: show image
  steps:
  - intent: show_image
  - action: show_image


- rule: start interaction
  steps:
  - intent: init_mapert
  - action: get_user_id
```

Appendix C

Paper on Entity Linking

Entity Linking in Tabular Data Needs the Right Attention

Miltiadis Marios Katsakioris
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, UK
mmk11@hw.ac.uk

Daniele Masato
Amazon Alexa
Cambridge, UK
masatod@amazon.co.uk

Yiwei Zhou
Amazon Alexa
Cambridge, UK
zhouyiwei1990@gmail.com

Abstract

Understanding the semantic meaning of tabular data requires Entity Linking (EL), in order to associate each cell value to a real-world entity in a Knowledge Base (KB). In this work, we focus on end-to-end solutions for EL on tabular data that do not rely on fact lookup in the target KB. Tabular data contains heterogeneous and sparse context, including column headers, cell values and table captions. We experiment with various models to generate a vector representation for each cell value to be linked. Our results show that it is critical to apply an attention mechanism as well as an attention mask, so that the model can only attend to the most relevant context and avoid information dilution. The most relevant context includes: same-row cells, same-column cells, headers and caption. Computational complexity, however, grows quadratically with the size of tabular data for such a complex model. We achieve constant memory usage by introducing a Tabular Entity Linking Lite model (TELL) that generates vector representation for a cell based only on its value, the table headers and the table caption. TELL achieves 80.8% accuracy on Wikipedia tables, which is only 0.1% lower than the state-of-the-art model with quadratic memory usage.

1 Introduction

Tabular data, such as web tables and databases, provides invaluable information about the world. According to Cafarella et al. (2018), by 2008, there are 14.1 billion HTML tables from Google's general-purpose web crawl, and 154M of them are high quality relational data. There has been various efforts to leverage rich factual information contained in tabular data for Knowledge Base Augmentation Ritze et al. (2016); Kruit et al. (2019), Question Answering Chen et al. (2021), etc. These applications require to automatically interpret and understand the semantic meaning of tabular data at scale. Entity Linking in tabular data, which target at linking a cell value ("Titanic") in tabular data with its corresponding real-world entity reference in a knowledge base (Q44578 in Wikidata[1]), is an important step for semantic table interpretation.

Comparing with Entity Linking in unstructured text Logeswaran et al. (2019); Martins et al. (2019), Entity Linking in tabular data needs to tackle some additional challenges. First, each cell value is an

[1] https://www.wikidata.org/

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

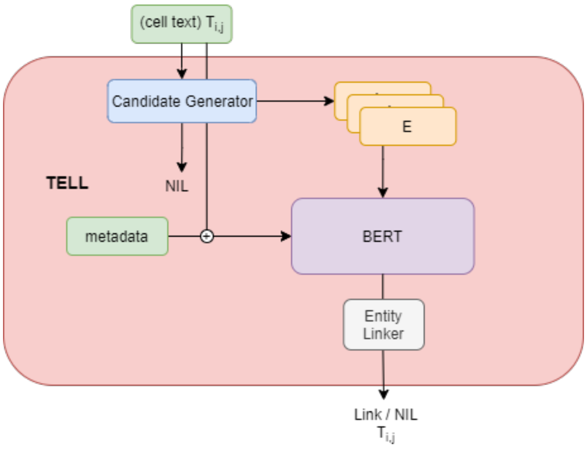


Figure 1: Overview of our EL pipeline. First, all entity mentions $T_{i,j}$ are passed through the candidate generator. The resulting list of candidates E and the $T_{i,j}$ with the corresponding metadata are encoded by BERT and the entity linker, ranks and makes the final decision for each $T_{i,j}$, whether it has a link from E or not.

entity mention to be linked, the environment it appears is not a complete sentence, but other entity mentions in short texts, real numbers, or dates. Second, besides content information in the form of rows and columns, tabular data is usually associated with metadata information, such as HTML page titles, table captions and column headers. For each cell in the table, this metadata information can contain both signals and noises. Third, sizes of tables can vary significantly. Once the table schema is determined, the size of a table can grow infinitely.

In this work, we focus on end-to-end Entity Linking solutions in tabular data, the main contributions are as follows:

- Besides entity disambiguation, we also consider the scenarios that a cell value is not an entity mention and the correct entity is not included in the list of candidate entities, so that the solution is robust to errors introduced during candidate entities retrieval.
- We reduce the required prior knowledge for existing entities in the targeted KB to entity names and entity descriptions, which increases the solution's generalisation capability to unseen entities.
- We experiment with various models to generate a vector representation for each cell value to be linked, and verified the importance of applying an attention mechanism and an attention mask to regulate the interactions between cell values and table metadata.
- We propose a simple yet effective model, Tabular Entity Linking Lite (TELL), which generates cells' vector representations only based on cell values and metadata. TELL reduces the computation complexity of tabular structure aware models from quadratic to linear while sacrificing the accuracy by 0.1%, from 80.9% to 80.8%.

2 Related Work

Most former works Ritze et al. (2016); Kruit et al. (2019) on Entity Linking in tabular data requires fact lookup during training and inference time. These solutions can only link cell values with entities that are well populated with facts in the targeted KB, and their entity linking capability is restricted to only cell values in the subject column of a table. There has been some learning based Entity Linking approaches for tabular data. However, they are either based on an assumption that the correct entity is included in the list of retrieved candidate entities, and only focus on the disambiguation part of the problem Deng et al. (2020); Luo et al. (2018); or they are dependent on some additional knowledge about the entities in the KB, such as prior probabilities of entity mentions Bhagavatula et al. (2015), entity embeddings Luo et al. (2018), or entity types Deng et al. (2020), which limit their generalisation capability to unseen entities during training.

3 Entity Linking in Tabular Data

Without loss of generality, we assume that each table T consists of M data rows and N data columns, and $T_{i,j}$ represents the cell value in the i^{th} row and j^{th} column. Each table can be associated with additional metadata, which includes: (1) Table caption C, a short text description summarizing the content of the table; (2) Page title P, the title of the web page the table was in; (3) Table headers $H = [h_0, \dots h_{N-1}]$, one for each column to define the table schema.

Same as Luo et al. (2018); Deng et al. (2020); Bhagavatula et al. (2015), we consider each table cell $T_{i,j}$ is a *potential* mention for real world entities in the target KB. Specifically, we define the Entity Linking in tabular data problem as follows:

Definition 1. Given a table T and a target KB, link cells $T_{i,j}$ in T with their corresponding real-world entity references in the target KB if possible, while automatically detect and ignore the ones that cannot be linked.

To increase the entity linking solution's generalisation capability to unseen entities Logeswaran et al. (2019) and any target KB, we only assume the existence of names and descriptions for entities in the target KB, which is minimal comparing with former works.

Figure 1 presents an overview of our EL pipeline which consists of two stages: (1) Candidate Generation, which generates candidate entities E for all cells in each table T, and (2) Entity Disambiguation, which ranks and selects the best $e \in E$ for each $T_{i,j}$ if there is any (refer to as NIL otherwise).

3.1 Candidate Generation

Most existing work Bhagavatula et al. (2015); Efthymiou et al. (2017); Ritze et al. (2015); Deng et al. (2020) directly use exact string matching-based lookup services provided by the target KB to generate candidate entities for each cell, which cannot tackle the variations of entity mentions in web tables. A more sophisticated method for this stage, we can increase the probability of including the right entity in the candidate entities (P_E) while introducing minimal noises. Following Ganea and Hofmann (2017), we used an gazetteer constructed based on Wikidata entity names and alias, as well as Wikipedia article titles, hyperlinks and redirects. Additionally, we applied BM25 to measure the similarity between a cell value and a KB entity rather than exact string matching. According to our analysis, the combination of gazetteer and BM25-based similarity search has increased P_E from 77% to 88%.

3.2 Entity Disambiguation

We use a shared BERT Devlin et al. (2019) encoder to encode all the textual inputs, which include: cell value $T_{i,j}$, metadata (C, P, H), entity name e_{name} and entity description e_{desc}.

For a KB candidate entity, its vector representation e is achieved by adding its averaged name token embedding and its averaged description token embedding:

$$e = mean(BERT(e_{name})) + mean(BERT(e_{desc})) \tag{1}$$

When encoding the cells of a table we have to take into account how to model the tabular structure, if and how to use metadata and the role of attention for the optimal representation. We compare different ways of encoding in a top-down approach. At the top, in terms of complexity and information load is $MaskAttEnc$, an encoder based on TURL Deng et al. (2020), a state-of-the-art framework for relational table understanding that consists of a structure-aware Transformer encoder to model the row-column structure of the table and capture the textual information and relational knowledge of each cell. During the self-attention calculation, a "hard-coded" attention mask limits the aggregation of information from one entity cell to other structurally related entity cells, such as cells in the same row/column. In our ablation study, when we are not using the hard-coded attention mask, we refer to the model as $AllAttEnc$.

We remove the hard-coded attention mask and we treat each cell value as a separate entity. Instead of encoding the metadata separately using extra attention heads and fusing them with the encoded cells, we concatenate them to each cell. We refer to the resulting module as TELL . $MaskAttEnc$

Splits	**Tables**	**NIL**	**Entities**
Train	554,239	16.6M	26.6M
Validation	4,738	142K	317K
Test	4,660	139K	312K

Table 1: Dataset statistics. In the 'NIL' column we show an estimate of the entity mentions that have no link and in the last column the total entity mentions $T_{i,j}$ from all tables.

makes predictions on the table level whereas TELL can make predictions on the individual cell level, allowing for extra flexibility.

We simplify the entity mention representation of TELL further by removing all BERT attention heads. We refer to these models as $SingleEnc$, with the cells being treated separately as in TELL. However, instead of attention, we encode the sentence embeddings either by an LSTM, $SingleLSTMEnc$ or by averaging the embeddings and passing them through a linear feedforward layer, $SingleLinearEnc$.

After getting the vector representation for a cell, we calculate the matching score between $T_{i,j}$ and e by,

$$P(e) = \frac{exp(T_{i,j} \cdot e)}{\sum_{e' \epsilon E} exp(T_{i,j} \cdot e')} \tag{2}$$

and we select the candidate with the highest probability.

4 Dataset

For our experiments we pre-processed and generated our own data splits of the WikiTables corpus Bhagavatula et al. (2015). WikiTable corpus contains 1.65M tables extracted from Wikipedia pages, and most of the tables contain hyperlinks between cell values and Wikipedia entity articles labelled by Wikipedia contributors.

We cleaned the table cells by lower-casing their content, removing HTML tags and removing special characters. For cells containing multiple hyperlinks, we retained the first link only. In order to fit training batches (25 batches) in memory, we discarded tables with more than 500 cells (entity mentions). We also discarded tables with no linked entities, and tables with either no candidates or more than 1800 candidates overall. For computational efficiency we remove duplicate cell values from the tables. In addition, to remove noises in Wikipedia hyperlinks, we compute the difference in lengths between each cell value and its corresponding linked entity's name. As long as the difference is bigger than 10, we will ignore the hyperlink because it is very unlikely to happen.

In order to map hyperlinked Wikipedia entities in WikiTables with Wikidata entities, we use Wikimapper[2].

Statistics of our data splits are summarized in Table 1. The table shows that the mean percentage of NIL entities in all splits is around 50%. The average number of NIL entities is consistent for both the dev and test splits.

5 Experimental Results

We evaluated our solution end-to-end using accuracy and F1 score. We first established a strong baseline, by encoding each cell, only with the cell value using BERT. This baseline, which does not use any extra context, achieves an accuracy of 77.5%. This baseline is simply learning mappings from cell value embeddings to entity embeddings, we expect any approach that uses additional context to outperform the baseline.

Table 2 shows increasing levels of ablation. It starts with $MaskAttEnc$, a model using both the whole table content and the metadata, then transitions to the aforementioned baseline model, a model using only separated cell values. The proposed solution, TELL , uses separated cell values as well as metadata, achieves 80.8% accuracy and 79.3% F1 score, and it only requires a linear number of parameters. The comparison between TELL and the baseline shows the importance of the metadata and the attention that happens in each cell between its text and the metadata.

[2] See `https://github.com/jcklie/wikimapper`.

Method	Acc	F1	Comp.
MaskAttEnc + meta (TURL)	**80.9**	**80.3**	$O(n^2)$
SingleAttEnc + meta (TELL)	*80.8*	*79.3*	$O(n)$
AllAttEnc + meta	78.6	78.1	$O(n^2)$
SingleLinearEnc + meta	77.9	76.1	$O(n)$
SingleAttEnc (baseline)	77.5	76.3	$O(n)$
AllAttEnc	76.8	76.0	$O(n^2)$
SingleLSTMEnc + meta	69.7	69.1	$O(n)$

Table 2: Evaluation of all variants on test set. In the Big O notation, n symbolises the number of input cells.

On the other hand, $MaskAttEnc$ attends to the whole table and achieves 80.9% accuracy with the help of the attention mask. This mask regulates that each cell only attend to structurally related cells from the entire table and the metadata. We see the importance of the attention mask when we compare it with $AllAttEnc + meta$ with a drop in accuracy of 2.3%. Without the attention mask as a regularization method, the model fails to automatically learn the most relevant information for each cell.

A marginal improvement (0.1%) in accuracy compared to TELL , for the sacrifice of scalability since computation complexity of the $MaskAttEnc$ solution grow exponentially $O(n^2)$. This is because $MaskAttEnc$ needs to model the relationship between any pair of cell values, rather than treating them separately as TELL. This also makes $MaskAttEnc$ able to only work on small tables. For TELL and its ablations, table size is not an issue, as long as the metadata are being passed on together with each cell. Aside from the type of attention and the structured relatedness, we see that the metadata are crucial and always contribute to the performance.

For tabular data is critical to apply the right attention in order to attend to the right cells when generating the cell representations. Otherwise, information dilution from irrelevant cells will greatly impact the performance and it might be better to simply pass each cell individually.

6 Conclusion

The state-of-the-art framework for relational table understanding (TURL) performs EL for a given cell using a Deep Neural Network with attention that aggregates information from the whole table, including the content and location of surrounding cells, and additional table metadata such as title, caption and headers. TURL achieves 80.9% accuracy on Wikipedia tables, but its computation complexity grows quadratically with the number of cells in a table.

In this paper we presented a lightweight approach for EL on tabular data that can achieve almost state-of-the-art accuracy with linear computation complexity. We consider both challenges of candidate retrieval and entity disambiguation, whilst trying to find a balance between the two. We showed that metadata are crucial. In order to avoid a noisy cell representation it is important to apply the right attention to avoid information dilution. In future work, we will focus on improving the candidate entity retrieval mechanism, by applying some embedding based approach. Another option is to develop solutions to transform tabular data into unstructured text, in order to leverage EL solutions for unstructured text to tackle the problem.

References

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Douglas C Downey. 2015. Tabel: Entity linking in web tables. In *ISWC*.

Michael Cafarella, Alon Halevy, Hongrae Lee, Jayant Madhavan, Cong Yu, Daisy Zhe Wang, and Eugene Wu. 2018. Ten years of webtables. In *VLDB*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2021. Open question answering over tables and text. In *ICLR*.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding through representation learning. In *VLDB*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Vasilis Efthymiou, Oktie Hassanzadeh, Mariano Rodriguez-Muro, and Vassilis Christophides. 2017. Matching web tables with knowledge base entities: From entity lookups to entity embeddings. In *ISWC*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *EMNLP*.

Benno Kruit, Peter Boncz, and Jacopo Urbani. 2019. Extracting novel facts from tables for knowledge graph completion. In *ISWC*.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *ACL*.

Xusheng Luo, Kangqi Luo, Xianyang Chen, and Kenny Zhu. 2018. Cross-lingual entity linking for web tables. In *AAAI*.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2019. Joint learning of named entity recognition and entity linking. In *ACL*.

Dominique Ritze, Oliver Lehmberg, and Christian Bizer. 2015. Matching html tables to dbpedia. In *WIMS*.

Dominique Ritze, Oliver Lehmberg, Yaser Oulabi, and Christian Bizer. 2016. Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *WWW*.