Pre-Confirmation Report

by

Sukai Huang

ORCID: 0000-0001-7886-5571

A pre-confirmation report for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Information Technology
School of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE

February 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, Deep Reinforcement Learning agents have outperformed human expert performance in complex video games, including Atari games and DOTA2 [Badia et al., 2020, Berner et al., 2019]. However, there are two significant issues among these Deep RL agents. The first is sample inefficiency: for instance, Hafner et al. [2020] required 10^8 episodes to surpass human performance when training their SOTA Atari RL agent DreamerV2. The second is low generalisation ability: an RL agent often requires a re-train process once the task environment is altered even a little bit. Berner et al. [2019] highlight this phenomenon in their report – every time DOTA2 updated its new patch, their agent had to re-train for days to adapt to new items and new hero skills even though the most of the game elements were unchanged.

The majority of people have no recollections of their first three to four years of life. For more than a century, psychologists have been baffled by this phenomenon known as "Childhood amnesia" [Eacott and Crawley, 1999]. Today, the mainstream belief among psychologists is that linguistic ability plays an important role in human memory as Peterson and Parsons [2005] suggested that preverbal memories are lost if children fail to describe them using their mother tongue language. It is thus reasonable to question whether an RL agent can improve sample efficiency and generalisation by combining with a natural language understanding module. Besides that, Natural-Language user interface (NLUI) is the most natural type of Human-Computer Interaction (HCI) interface. Therefore, Natural Language Understanding of Learning agents is also an active area of study in both Natural Language Processing (NLP) and Reinforcement Learning (RL).

Luketina et al. [2019] described the study as "Reinforcement Learning informed by Natural Language". Meanwhile, in the field of NLP, researchers are also aware of the great potential in using interactive environments to boost natural language understanding models – namely "Grounded Language Learning (GLL)" – which studies how to understand a word through interactive environments instead of the company context it keeps. Both studies ended up trying to solve the same research tasks even though their motivations are different.

Previous efforts have been put on two directions: (1). introducing various training environments that are suitable for grounded language learning [Kiseleva et al., 2021, Côté et al., 2018, Shridhar et al., 2020, Chevalier-Boisvert et al., 2018, Zhong et al., 2021, Narayan-Chen et al., 2019, Hill et al., 2020]; (2). designing various NLP-RL models that can be categorised into three forms: (a). Multi-modality fusion model [Cao et al., 2020, Bianchi et al., 2021, Mao et al., 2021] (b). Instructor-Executor model [Andreas et al., 2017, Kiseleva et al., 2021, Jiang et al., 2019, Hu et al., 2019] (c). Reward shaping model [Goyal et al., 2019, 2020] (See Figure 1.1). Nevertheless, these

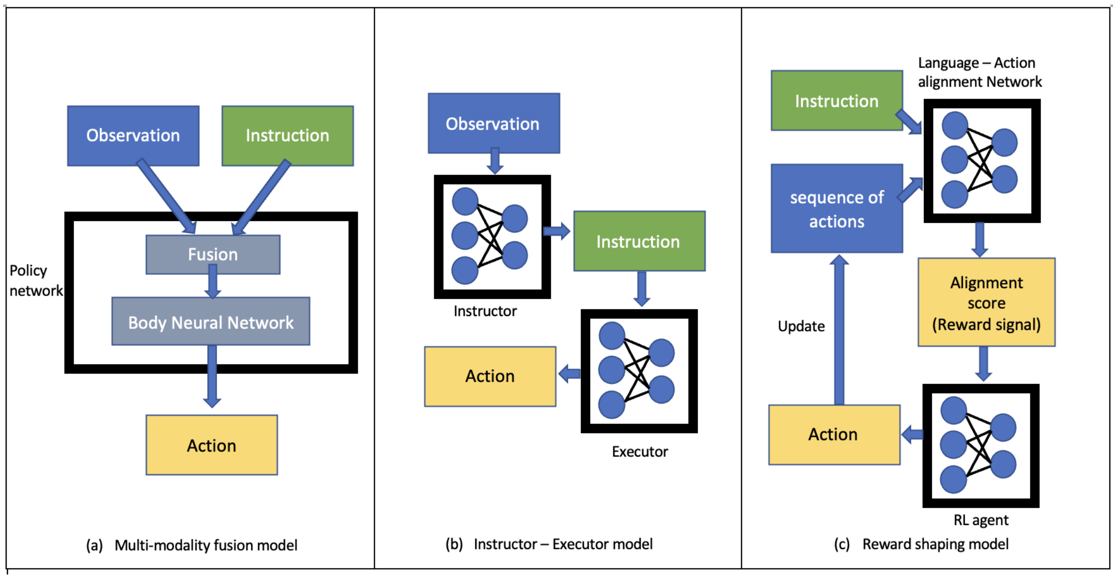


FIGURE 1.1: Illustrations of three different types of NLP-RL models. (a) Multi-modality fusion model treat both observations from the environment and the language instructions as data input with different modality. Therefore, the model will first transform them into the same latent space, which is simialr to ViT model [Dosovitskiy et al., 2020]. (b) Instructor – Executor model firstly convert observations into corresponding instructions using Instructor sub modules, after that the intermediate instructions will be fed into Executor sub module, thereby returning actions that follow the instruction. (c) Reward shaping model provided RL agents with an additional reward signal that is based on to what extent the agent trajectory is aligned with the instruction.

existing models have failed to explicitly address the characteristics of natural language expressions, including compositionality, levels of abstraction, anaphora, ellipsis, bridging and more (See Table 1.1). Disregarding these characteristics prevents NLP-RL models

Characteristics	Description	Example
Compositionality	The meaning of a complex natural language expression is determined by the meanings of its constituent expressions and the rules used to combine them.	An instruction could be composed of several sub-goals that require RL agents to act accordingly.
Levels of Abstraction [Rasmussen, 1986]	The idea of removing certain details or attributes so as to focus attention on details of greater importance.	E.g., for Angry Birds game, a high level strategy could be: "targeting weak points to break the stability of a structure", a low level command could be: "aiming 36.7 degrees with 0.7 unit of force."
Anaphora	The use of a word such as a pronoun that has the same reference as a word previously used in the same discourse.	"John wrote the essay in the library but Peter did **it** at home."
Ellipsis	Leaving out words rather than repeating them unnecessarily.	"I want to go but I can't" instead of "I want to go but I can't **go**."
Bridging	In addition to signalling what the new contents are about, bridging words connect new and old paragraphs with some sort of relationship.	"Phillip, the son, becomes very angry, **while** her daughter Karin is more indulgent."

Table 1.1: Example of characteristics of natural language expression, which are often not addressed by current NLP-RL models

from performing well in task of interactive grounded language learning. For instance, no NLP-RL models have yet successfully reconstructed the whole buildings by following the instructions in IGLU Silent Builder competition.

Therefore, the focus of my research is to better address the characteristics of natural language for NLP-RL model so as to improve the performance in the task of grounded language learning (GLL). Specifically, we concentrate our efforts on one subproblem inside GLL: mapping sequences of executable representations to natural language utterances (i.e., semantic parsing).

1.2 Objectives

This research studies how a NLP-RL model can explicitly address the characteristics of natural language expression in the interactive grounded language learning task. The main hypothesis in this research is the following:

> *The ability to explicitly address the characteristics of natural language expressions can lead to better mapping from natural language instructions to logical sequences of actions, and thus improve NLP-RL model performance in the grounded language learning task*

To this end, the following research questions are investigated:

1. **How can we effectively handle "abstraction hierarchy" of unstructured natural language instructions explicitly?** Specifically, natural language information can be distilled into two dimensions – "sequential composition" and "hierarchical composition", as shown in Figure 1.2.

2. **In term of "sequential composition", how can we improve the alignment between natural language utterances and the agent's trajectory actions?** In an interactive learning environment that involves a Builder agent and an Architect agent (e.g., works from Jayannavar et al. [2020], Jernite et al. [2019]), the Architect needs to comprehend the behaviours of Builder agents so as to give the following commands. For this reason, it must summarize the trajectory of agents and encode the information using natural language. We may encounter situations where a sequence of actions could correspond to a group of goals, or two sequences of actions with different order and length could refer to a same goal. That's what brought alignment ability, a necessity of comprehending agent's trajectories with varying lengths and a prerequisite for compositional generalization (See Figure 1.3).

3. More research questions will be investigated as we delve deeper into the interactive grounded language learning in a collaborative environment. As an example, IGLU contest asked for an Architect agent that can deliver precise instructions to the Builder and also accurately respond to enquiries. Thus, we can also investigate **how to establishing a common ground on the degree of abstraction so as to reduce uncertainty when Architect and Builder agents communicate.**

Although this research focuses on grounded language learning, we expect the contributions will also apply to other multi-modal learning problems such as visual question

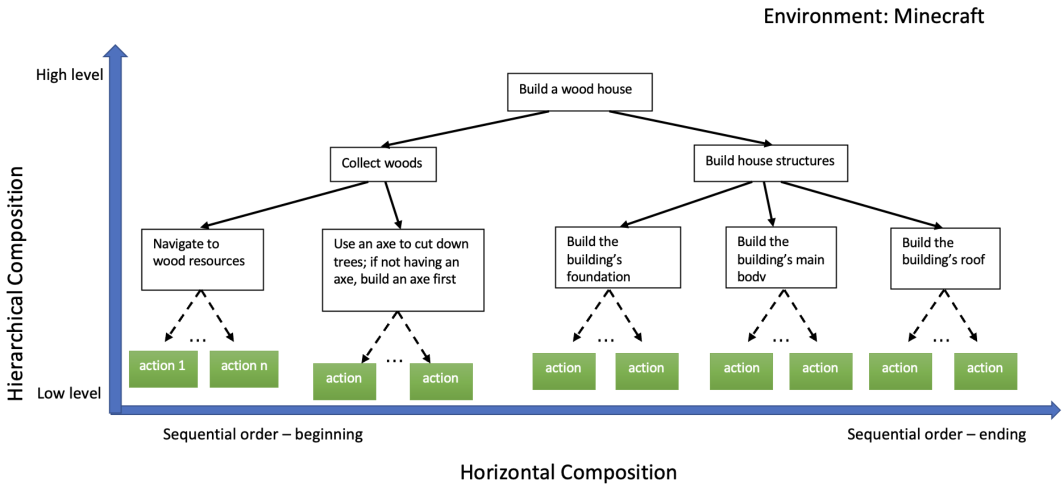
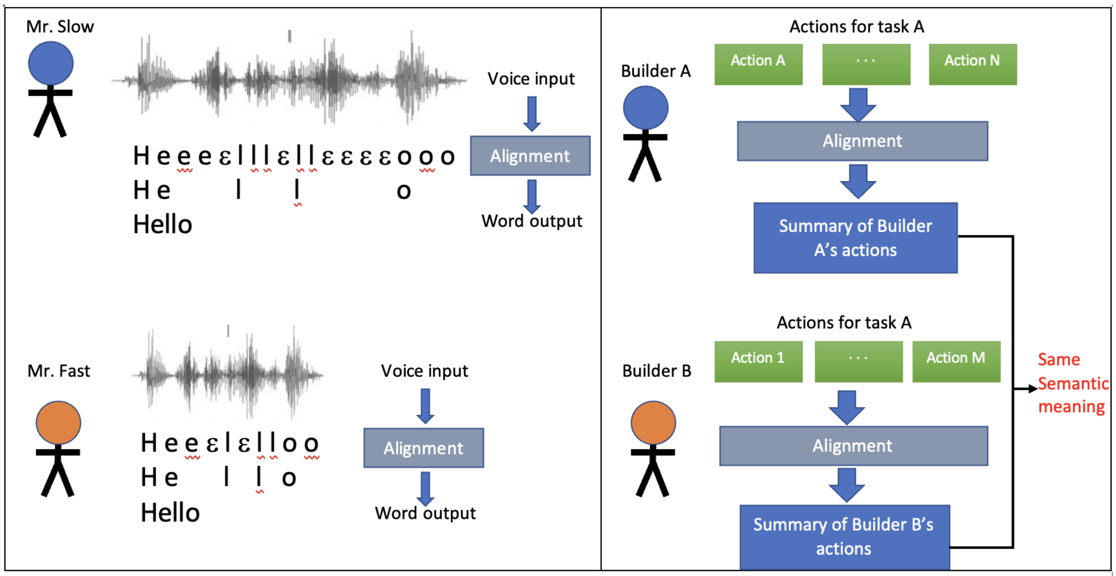


FIGURE 1.2: An example of "sequential composition" and "hierarchical composition" in natural language instructions. "Hierarchical abstraction": when humans communicate, they would assume a shared knowledge foundation and hide the low-level details in order to focus attention on details of greater importance. Hence, a RL agent needs to deduce corresponding low level details from high level instructions so that concrete actions can be derived; "Sequential composition": events were not guaranteed to be described in sequential order. Thus, a RL agent must recognise pre- and post-conditions of events and arrange them correctly.

answering (VQA), image and video captioning, etc. Grounded language learning attracts my interests more due to the interactive nature of its learning environment.



(a) Speech recognition alignment

(b) Natural language generation alignment (the reverse of semantic parsing)

FIGURE 1.3: Comparison between temporal alignment in speech recognition and alignment in grounded language learning. A well designed alignment method helps different input sequences with the same semantic meaning to converge to the same output regardless of the length of the input sequence.

Chapter 2

Research Question

2.1 How can we effectively handle "abstraction hierarchy" of unstructured natural language instructions explicitly?

Humans communicate by assuming a shared knowledge base and hide the low-level details in order to focus attention on details of greater importance. So, if a task is asking for building a wood house. A human instructor would only say "First of all, collect woods. After that, use the woods to build the building structure" and will not give details about how to collect woods and build structures if one assumes that the player knows it already (Shown in Figure 1.2). The advantages of hiding details were not only an increase of communication efficiency (as the information is more condensed), but also an increase of generalisation performance because the information becomes more task-independent. For example, The exact low level operations for "cutting trees" are different in Minecraft and Warcraft. For this reason, Intelligent agents should comprehend and communicate a concept over a certain level of abstraction and leave the low level details to task-specific modules. Interestingly, when playing games, we, the human participants, are strong at comprehending high-level strategies but lousy at executing a sequence of low-level commands precisely. For example, when playing Angry Birds game, a high-level strategy is simple for human players to follow, yet instructions like "shot at 36.7 degrees with 0.7 unit of force" are difficult to execute. However, the situation is the opposite for AI agents – they are capable of converting low-level language instructions into the corresponding logical forms and then execute accordingly, but are incapable of comprehending high level strategies that often looks straightforward for humans (See Table).

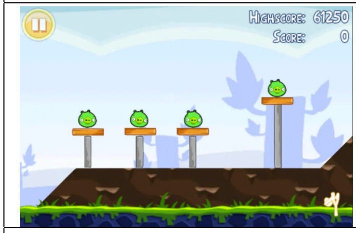
Physics Simulation Game	High Level Strategy	Low Level Commands
	First fire Red over all the pigs, so he bounces back off the slanted wall on the far side. ...	• Aim at 36.7 degrees with 0.7 unit of force • ...
	The Sport Car's bridge must arc upwards, and the Van's bridge must arch downwards. ...	• Place wood road at (X,Y) coordinate. • ...

TABLE 2.1: Physics Simulation Game examples with strategies at different levels of abstraction. The first game is Angry Birds and the second game is Poly Bridge 2

To address "abstraction hierarchy" issue for NLP-RL agents, the following research question(s) will be investigated.

2.1.1 How feasible can we extend the modular multitask reinforcement learning from "structured policy sketches" to "unstructured natural language instructions"?

We explain the terminology first before going into details about the research question.

Multitask RL environment problem arise from a group of infinite-horizon Markov decision process in a shared environment. This environment is specified by a typle $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma)$, with $\mathcal{S}$ a set of states, $\mathcal{A}$ a set of concrete executable actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ a transition probability distribution, and γ a discount factor. Each task $\tau \in \mathcal{T}$ is then specified by a pair (R_τ, ρ_τ), with $R_\tau : \mathcal{S} \to \mathbb{R}$ a task-specific reward function and $\rho_\tau : \mathcal{S} \to \mathbb{R}$ an initial distribution over states. Angry Birds is an example of a multitask RL environment problem because the RL agent must solve a family of game levels while characters and items are shared across levels.

Option MDP model was introduced by Sutton et al. [1999], which solve a RL problem by maintaining a set of sub-policies $\{\pi_0 ... \pi_i\}$. A policy π_i in Option MDP model is a mapping from state $\mathcal{S}$ to actions plus a stop signal $\mathcal{A} \cup \{\texttt{STOP}\}$. A sub-policy network will executes actions until the `STOP` symbol is emitted, at which point control is passed to the next sub-policy π_{i+1}

Policy Sketch is defined by Andreas et al. [2017] as short, ungrounded, symbolic representations of a task that describe its component parts. For example, the task "make planks" is described as a sequence of sketches "get wood, use workbench" (See Figure 2.1). In the paper the policy sketches of a task is setup by hand and can be shared across different tasks. The term "policy sketch" is also used in classical search and planning area [Drexler et al., 2021, Bonet and Geffner, 2020]. It is a set of policy rules that is designed by hand and able to characterise a problem. The key impact of the use of policy sketch is the decomposition of complex tasks, thereby reducing the complexity of the task and achieving compositional generalisation.

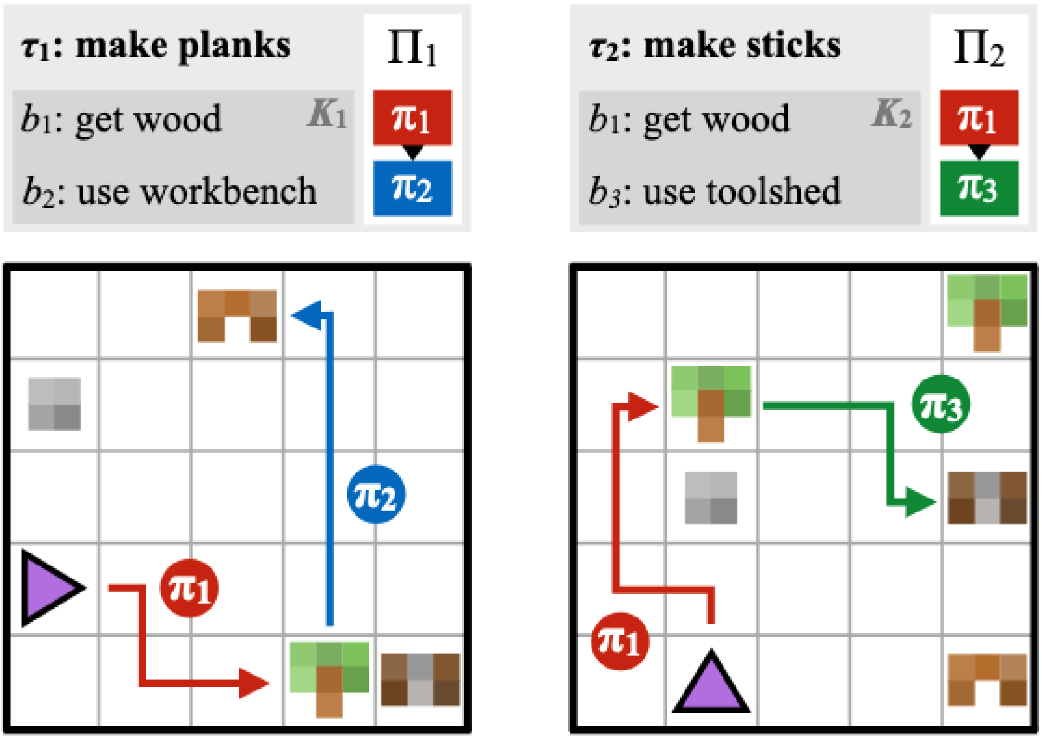


FIGURE 2.1: Credit by Andreas et al. [2017], learning from policy sketches. The figure shows simplified versions of two tasks (*make planks* and *make sticks*), each associated with its own policy (Π_1 and Π_2 respectively). These policies share an sketch b_1 (*get wood*): both require the agent to get wood before taking it to an appropriate crafting station. In this work, each sketch b_i will be assigned with a sub-policy π_i. The reusable sub-policies are hence provide compositional generalisation across multitasks.

Previous work from Andreas et al. [2017] has shown that in a "multitask reinforcement learning environment", "options MDP model" that is guided by handmade "policy sketch" achieve better performance than existing common singular RL models. Furthermore, the model has better zero-shot generalisation performance as an unseen task can be treated as an integration of familiar sub-tasks and thus can be handled by reusable sub-policies.

The Option MDP model provides an explicit way to address the "abstraction hierarchy" feature of natural language. It can be viewed as a hierarchical RL model because the sub-policy networks can be assigned to different hierarchical levels and each only focuses on parts of a task at a exact level of abstraction. Therefore, we plan to extend Andreas et al. [2017]'s approach by adapting the modular multitask reinforcement learning from "structured policy sketches" to "unstructured natural language instructions".

Their delimitation to break:

When we consider unstructured natural language instruction, the number of vocabulary in the corpora will be largely increased. In their work, they tried to keep a small size of vocabulary otherwise they need to maintain numerous policy networks (See Figure 2.2)

Goal	Sketch				
Crafting environment					
make plank	get wood	use toolshed			
make stick	get wood	use workbench			
make cloth	get grass	use factory			
make rope	get grass	use toolshed			
make bridge	get iron	get wood	use factory		
make bed*	get wood	use toolshed	get grass	use workbench	
make axe*	get wood	use workbench	get iron	use toolshed	
make shears	get wood	use workbench	get iron	use workbench	
get gold	get iron	get wood	use factory	use bridge	
get gem	get wood	use workbench	get iron	use toolshed	use axe
Maze environment					
room 1	left	left			
room 2	left	down			
room 3	right	down			
room 4	up	left			
room 5	up	right			
room 6	up	right	up		
room 7	down	right	up		
room 8	left	left	down		
room 9	right	down	down		
room 10	left	up	right		

FIGURE 2.2: Credit by Andreas et al. [2017], In their experiments, they keep a very small size of vocabulary and therefore they can maintain a feasible number of subpolicy networks, the advantages are: (1). ensuring enough training for each policy network (2). ensuring enough knowledge sharing among different tasks.

The number of words will surge in terms of unstructured natural language instructions. Additionally, the model must also know how to deal with synonyms, paraphrases, and so on.

But one delimitation is reasonable, which is to keep a small number of policy networks. For this reason, we may want to **cluster** the unstructured natural language instructions and assign each policy network to one cluster.

Their assumptions to break: In the previous work, in order to provide a smooth learning experience, they selected simple tasks to train first (i.e. curriculum learning). Their assumption was: a brand new RL agent cannot solve complex tasks as they can easily get stuck in local optimum. Therefore, if they provide it with simple tasks first, the agent can gain some basic knowledge about the environment and then it can continue to learn more complex tasks smoothly. However, they assume that task with smaller length of policy sketch is considered as "simple". e.g., see the figure above, "make rope" is simpler than "make bridge" due to a smaller length of the sketch instruction.

This assumption is broken when we consider unstructured natural language instructions. For example, "building house" is not simpler than "moving towards north for two steps". Instead, the agent must understand the level of abstraction of the instructions and recognise how dense the information is. Therefore, a new algorithm that can rate the difficulty of tasks based on the unstructured natural language instructions is required.

To test our new settings, we will use IGLU builder contest as the research environment. IGLU contest contained 150+ building tasks and provided grounded human to human dialogues data for training. Hence, we are able to obtain both a multitask RL environment and unstructured natural language instructions. We will also consider tricks that can accelerate the training process such as curriculum learning and hindsight instruction relabelling [Andrychowicz et al., 2017]. More details will be discussed in Chapter 3

2.2 In term of "sequential composition", how can we improve the alignment between natural language utterances and the agent's trajectory actions?

In natural language instructions, events were not guaranteed to be described in a sequential order. For example, the instruction "Before you **cut trees**, you should **get an axe**" indicates that an agent may perform actions that achieved the goal of "getting an axe" in the first place, even though the phrase "get an axe" appears after the phrase "cut trees". Moreover, the sequential composition problem may become more complicated as an mentioned event can be repeated and mixed up with others. For example, an instruction "Drive a car to City A, refuel if you run out of fuel" would expect that refuel event took place **during** the driving event instead of after, and also repeat refuel event once the fuel is used up. As shown above, the order of composition is not explicitly stated in natural language instructions, which leads to uncertainty.

In this research, we focus on the actions to text summarization problem – which can also be treated as a task of alignment between natural language utterance and the associated sequence of actions, without aligned training data provided. Specifically, given a sequence of action $\{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n\}$ and a sentence of corresponding natural language instruction $\{w_1, w_2, ..., w_m\}$ where $\mathcal{A}$ is a concrete low level action and w is a word and $m \leq n$, an event e_i can be represented as a part of consecutive words

($e_i = f_1(\{w_i, ..., w_{i+j}\})$) where f_1 is the mapping from words to the associated event. Similarly, we have semantic mapping from actions to event $e_i = f_2(\{\mathcal{A}_i, ..., \mathcal{A}_{i+j}\})$. The alignment task is to find a mapping h such that $\{\mathcal{A}_i, ..., \mathcal{A}_{i+j}\} = h(\{w_i, ..., w_{i+j}\})$ where $h = f_2^{-1} \cdot f_1$.

Specifically, the following research question(s) will be investigated.

2.2.1 How feasible can we extend Connectionist Temporal Classification (CTC) operation from automatic speech recognition to actions to text summarization?

Background:

An architect agent must comprehend builder agents' external action trajectories in case where the architect had no access to builders' internal policies. In addition, for an architect with a natural language user interface, it is also necessary to summarise the builder's behaviour and encode it in natural language. As semantic parsing converts natural language utterances to logical forms, the action to text summarization task is more like a reverse semantic parsing task as it tries to convert logical forms back to natural language. The reverse semantic parsing process was often used as a part of dual learning in order to assist semantic parsing training [Cao et al., 2019]. In this study, we took inspiration from automatic speech recognition, which also requires mapping from tokens with varied lengths to labels. Specifically, out study focuses on the "alignment" problem, which involves aligning the builder's actions to NL utterances with no aligned training data provided.

There are challenges that prevent us from using simpler supervised learning techniques. In particular:

1. An action sequence may contribute to a group of aims, thus both input sequences and output sequences can vary in length.
2. There is no precise alignment training labels between input sequences and output sequences (correspondence of the elements).

The alignment process for automatic speech recognition (ASR) presents similar challenges. (See Figure 2.3). We propose that, because the alignment problem in ASR can be largely solved by Connectionist Temporal Classification (CTC) operation [Graves et al., 2006], we investigate the feasibility of extending CTC to actions to text summarization.

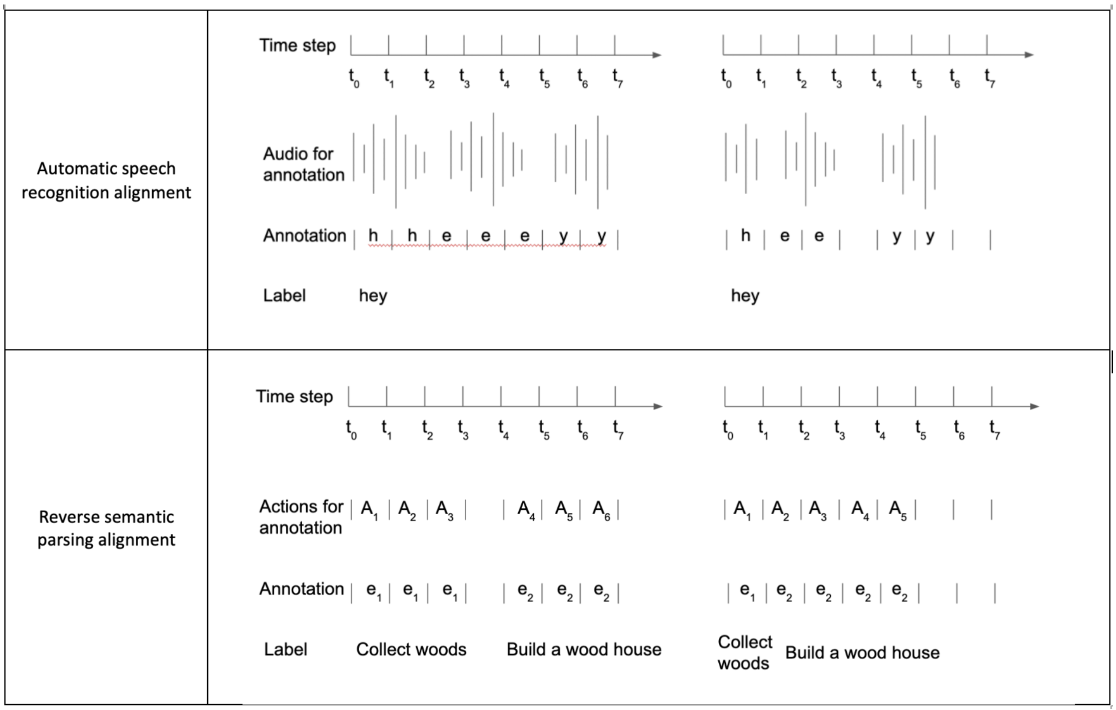


FIGURE 2.3: Actions to text summarization and automatic speech recognition tasks are similar in a way that both require alignment from input sequences with varied lengths to a group of labels. In both cases, different input sequences would result in the same output since they are semantically similar.

Delimitations of the study:

Recall that CTC alignment possesses a few properties:

1. The alignments between input sequences and output sequences are monotonic. Specifically, imagine there is a pointer on the output sequence; as we advance to the next input, the pointer can only either stay put or advance one step. The monotonicity property is a prerequisite for dynamic programming (DP), the core algorithm of calculating the loss value in CTC.
2. The alignment of input sequences to label sequences is many to one. In ASR, one atomic input unit of audio signal can only have one label. Similarly, in GLL, one atomic low-level action can only contribute to one sub-goal.
3. We may infer a third property from the preceding two: the length of input sequences must be greater than that of output labels (See Figure 2.4).

Therefore, in order to preserve the property of CTC, our project only analyses cases in which the length of action sequences exceeds that of natural language utterances. As such, the task is treated as a special summarization task so that the size output utterances is short. As it may be observed, this upside down version is more likely to

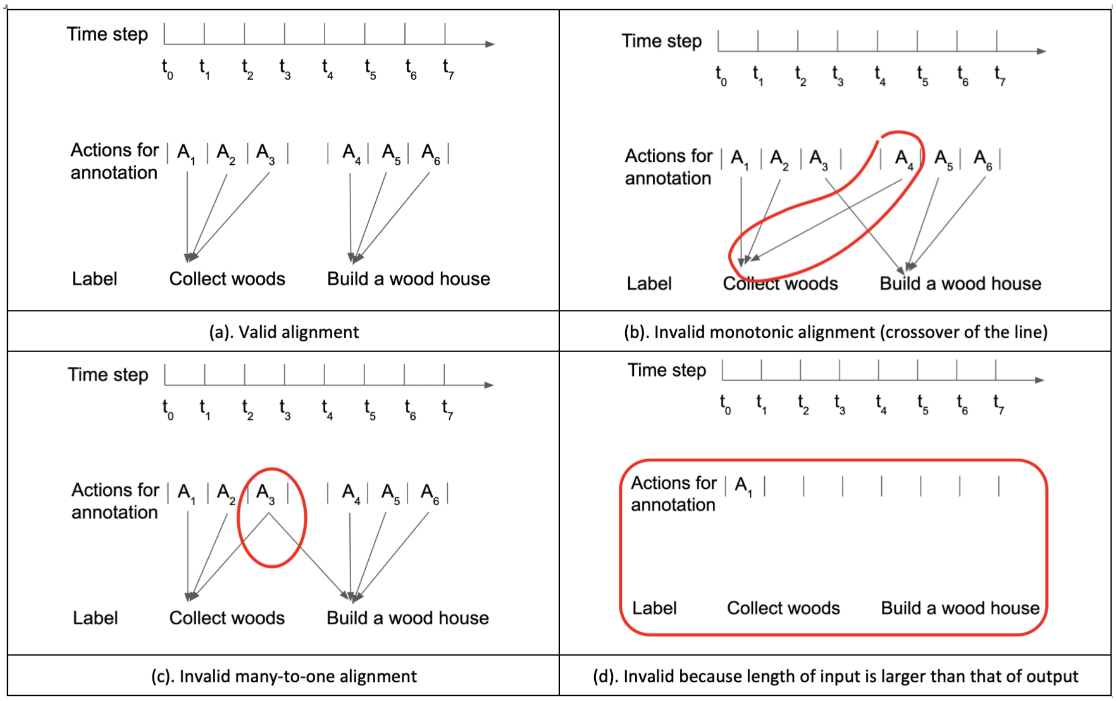


FIGURE 2.4: An illustration of a valid and invalid alignments for Connectionist Temporal Classification operation.

preserve the properties than normal semantic parsing because an agent's trajectory can be discretized into a larger number of units than natural language sentences. Besides, we assume that both input and output sequences are temporally coherent to a certain extent (i.e. no events are repeated and mixed up) so as to preserve monotonicity.

2.3 More research questions

More research questions are about to come as we further break the delimitations and assumptions. For instance:

- How to establishing a common ground on the degree of abstraction so as to reduce uncertainty when Architect and Builder agents communicates.
- How can we go beyond CTC properties so as to comprehend repeated events or mixture of events?

Chapter 3

Research Progress

IGLU (https://www.iglu-contest.net/) is our main research environment. Currently it is holding "Silent Builder competition", which is a suitable research environment for our research on Modular NLP-RL model (See section 2.1.1). In the future (probably after midyear 2022), the organisation will launch new "Architect competition", which will provide a suitable environment for our research on summarization alignment problem (See section 2.2.1).

To date (6 Feb 2022), we are working on building a baseline model for Silent Builder of IGLU competition. The baseline model will serve as a basic framework for future research and development. This step is necessary because NLP-RL model is an integration of multiple sub-modules that are responsible for multi-modal data, ranging from visual observations, symbolic state records to natural language instructions. Therefore, besides the NLP part, we must build other modules too for a NLP-RL agent to function.

Details of the baseline model can be viewed in my weekly progress report pages:

1. https://sino-huang.github.io/docs/phd-weekly-progress/19-dec-25-dec-2021/
2. https://sino-huang.github.io/docs/phd-weekly-progress/2-jan-8-jan-2022/
3. https://sino-huang.github.io/docs/phd-weekly-progress/9-jan-15-jan-2022/
4. https://sino-huang.github.io/docs/phd-weekly-progress/16-jan-22-jan-2022/
5. https://sino-huang.github.io/docs/phd-weekly-progress/23-jan-29-jan-2022/

3.1 Expectation

1. By 20 Feb 2022, complete a baseline Builder model and submit to the competition organisation

2. By Jun 2022 (Not Confirmed), finish modular NLP-RL research 2.1.1

3. By Nov 2022 (Not Confirmed), finish summarization alignment research 2.2.1

Bibliography

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning*, pages 166–175. PMLR, 2017.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, pages 507–517. PMLR, 2020.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dkebiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

MJ Eacott and RA Crawley. Childhood amnesia: On answering questions about very early life events. *Memory*, 7(3):279–292, 1999.

Carole Peterson and Brenda Parsons. Interviewing former 1-and 2-year olds about medical emergencies 5 years later. *Law and Human Behavior*, 29(6):743–754, 2005.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. *arXiv preprint arXiv:1906.03926*, 2019.

Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, et al. Neurips 2021 competition iglu: Interactive grounded language understanding in a collaborative environment. *arXiv preprint arXiv:2110.06536*, 2021.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer, 2018.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*, 2018.

Victor Zhong, Austin Hanjie, Sida Wang, Karthik Narasimhan, and Luke Zettlemoyer. Silg: The multi-domain symbolic interactive language grounding benchmark. *Advances in Neural Information Processing Systems*, 34, 2021.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. Collaborative dialogue in minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, 2019.

Felix Hill, Olivier Tieleman, Tamara von Glehn, Nathaniel Wong, Hamza Merzic, and Stephen Clark. Grounded language learning fast and slow. *arXiv preprint arXiv:2009.01719*, 2020.

Tianshi Cao, Jingkang Wang, Yining Zhang, and Sivabalan Manivasagam. Babyai++: Towards grounded-language learning beyond memorization. *arXiv preprint arXiv:2004.07200*, 2020.

Federico Bianchi, Ciro Greco, and Jacopo Tagliabue. Language in a (search) box: Grounding language learning in real-world human-machine interaction. *arXiv preprint arXiv:2104.08874*, 2021.

Jiayuan Mao, Haoyue Shi, Jiajun Wu, Roger Levy, and Josh Tenenbaum. Grammar-based grounded lexicon learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Yiding Jiang, Shixiang Gu, Kevin Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *arXiv preprint arXiv:1906.07343*, 2019.

Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. Hierarchical decision making by generating and following natural language instructions. *arXiv preprint arXiv:1906.00744*, 2019.

Prasoon Goyal, Scott Niekum, and Raymond J Mooney. Using natural language for reward shaping in reinforcement learning. *arXiv preprint arXiv:1903.02020*, 2019.

Prasoon Goyal, Scott Niekum, and Raymond J Mooney. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards. *arXiv preprint arXiv:2007.15543*, 2020.

Jens Rasmussen. Information processing and human-machine interaction. *An approach to cognitive engineering*, 1986.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. Learning to execute instructions in a minecraft dialogue. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 2589–2602, 2020.

Yacine Jernite, Kavya Srinet, Jonathan Gray, and Arthur Szlam. Craftassist instruction parsing: semantic parsing for a minecraft assistant. *arXiv preprint arXiv:1905.01978*, 2019.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Dominik Drexler, Jendrik Seipp, and Hector Geffner. Expressing and exploiting the common subgoal structure of classical planning domains using sketches: Extended version. *arXiv preprint arXiv:2105.04250*, 2021.

Blai Bonet and Hector Geffner. General policies, serializations, and planning width. *arXiv preprint arXiv:2012.08033*, 2020.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Ruisheng Cao, Su Zhu, Chen Liu, Jieyu Li, and Kai Yu. Semantic parsing with dual learning. *arXiv preprint arXiv:1907.05343*, 2019.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.