Integrating LLMs into the DIARC Cognitive Architecture to Resolve Past Temporal References

Valerio Farriciello and Frank Förster
School of Physics Engineering and Computer Science
University of Hertfordshire
valerio.farriciello@gmail.com f.foerster@herts.ac.uk

Abstract

We report on our efforts to resolve past temporal references – past temporal deictic and past discourse deictic expressions – as well as pronominal anaphora[1], present in instructions given to robots controlled by the DIARC cognitive architecture. Instructions given to a robot, are sent to a large language model (LLM) to firstly determine whether these contain such references, and, in the affirmative case, to rewrite the instructions such that they comply with the format accepted by the parser of DIARC's natural language understanding component (NLU). The preliminary results are promising.

1 Introduction

Despite considerable progress of LLMs and the related Large Reasoning Models (LRMs) in recent years, these models still suffer from problems such as hallucinations (Ji et al., 2023), limited reasoning capabilities (Shojaee et al., 2025), and unwarranted confidence in their knowledge (Yin et al., 2023). This renders their use as exclusive high-level controllers for robots problematic. By contrast, Cognitive Architectures (CAs), especially the symbolic types, do not suffer from these problems, but will typically only allow limited and prescribed forms of natural language instructions. Authors such as Sun (2024) and Romero et al. (2023) therefore suggested to integrate LLMs with CAs to obtain the "best of both worlds": robustness and reliability in terms of reasoning and planning, as well as flexibility in terms of language instructions. Prior work on reference resolution in DIARC (Scheutz et al., 2018) focused on spatial references, developing distributed open-world mechanisms for grounding spatial references such as "the room across from the kitchen", and combining those mechanisms with dialogue strategies that ask for clarification when a description matches more than one place (Williams and Scheutz, 2016, 2017). Our efforts extend this line of work by focusing on temporal references.

[1]For simplicity, we will in the following refer to this trio simply as '(relevant) deictic expressions'

2 Methods

Given DIARC's component-based architecture we chose to integrate the LLM as a new module of the CA (cf. Romero et al. 2023): the *PastReferenceComponent* (*PRC*). Claude 3.5 Sonnet was chosen as LLM, mainly due to its large context window of 200000 tokens which allows for the processing of extended dialogue histories. Due to limitations in terms of computational power, Claude was not executed locally, but a cloud-based instantiation was used instead.

Integration Despite DIARC's modularity, its NLU component needed to be modified slightly to create a *dialogue history* and inject calls to the *PRC* which requires this history for resolving references. The *dialogue history* stores all valid user and robot utterances with a timestamp and unique index per utterance for each user session.

Processing of Instructions & Reference Resolution Loosely following the *Manual Chain of Thought* approach (Zhang et al., 2022), the PRC's processing logic for handling utterances with potential deictic expressions was divided into three steps (see appendix A for more details)

Step 1 – Identification – determines whether an utterance contains a relevant deictic expression. The PRC first checks whether a dialogue history exists. If one exists, a system prompt is constructed containing an explanation of the target concepts: past temporal and discourse deixis and pronominal anaphora. The system prompt further contains instructions to analyse the utterance solely based on what is provided and that the response should consist of a simple *yes* or *no*. Optionally, extra context can be provided (see below). If the LLM's

answer is *yes*, the PRC continues with step 2.

Step 2 – Referencing – has the objective to identify the past utterance containing the referent that a deictic expression or anaphor refers to. Another system prompt is constructed consisting of the indexed dialogue history, including speaker roles, an instruction to analyse this history with a view to identify any previous statement that might be referred to by the utterance, and return a response in a prescribed format including a short explanation.

Step 3 – Rephrasing – aims to transform the user utterance into a contextually complete and unambiguous command that DIARC can understand and act upon. A new system prompt is created consisting of (i) the past utterance containing the referent or antecedent as determined in step 1 (ii) the explanation from step 1, (iii) a list of valid words extracted from the DIARC dictionaries, (iv) a list of previously generated invalid inputs. The prompt is further extended, instructing the LLM to generate a single, clear, and concise phrase that incorporates the referent and some further instructions.

Finally, the resulting paraphrase is validated. If invalid, the paraphrase is added to the list of invalid inputs and step 3 is repeated. If valid, it is returned to DIARC's default NLU pipeline. After at most three unsuccessful rephrasing attempts the PRC gives up and returns the original utterance to DIARC's NLU pipeline.

Extra Context: Preliminary testing of the PRC indicated that the addition of some situational context to the system prompts improved the odds of detecting and rephrasing deictic expressions. The context consisted of the description "In this scenario, there are 4 actors: 2 are robots called Shafer and Dempster, and 2 are humans called Evan and Ravenna who give instructions to the 2 robots".

Evaluation The PRC module was evaluated using the simulation *TwoNaoDemo* (Scheutz et al., 2024), in which two robots (Dempster and Shafer) interact through natural language with two human interlocutors (Evan and Ravenna) to perform simple collaborative tasks in a shared environment. Twelve mini dialogues were designed - four each targeting past temporal deixis, past discourse deixis, and pronominal anaphora. These dialogues can be found in the appendix A.
Three tests were performed: 1) using DIARC as is without integrated PRC ("pre-test"), 2) using DIARC with integrated PRC, but without using extra context, and 3) using DIARC with integrated PRC and extra context.

3 Results

The test yielded the results shown in Table 1.

	Pre-Test	**Test 1**	**Test 2**
Past Temporal Deixis (PTD)	0/4	1/4	3/4[1]
Past Dialogue Deixis (PDD)	0/4	2/4[2]	3/4[3]
Pronomial Anaphora (PA)	0/4	1/4[4]	4/4

Table 1: Success rates of the PRC module without (Test 1) and with extra context (Test 2) in paraphrasing expressions containing deictic expressions of the stated type. x/y: x successful tests (out of y). Numbers in brackets refer to additional notes on failures in the main text.

[1][2] Here, the PRC produces a correct circumscription of the relevant utterance, but the dialogue fails due to some NLU error downstream.
[2] One dialogue fails due to the PRC not detecting a PDD utterance ("false negative"), a second one fails due to it incorrectly flagging an utterance up as PDD-containing ("false positive").
[3] The failed test here is due to the same false positive as in [2].
[4] One false positive, one false negative, and one error due to a failure in identifying the addressee correctly, but with an otherwise correct paraphrase.

4 Discussion, Conclusion & Future Work

The success rate of the PRC without extra context is moderate in paraphrasing utterances with deictic expressions (50%), especially with respect to PTD utterances (25%). Adding extra situational context to the system prompt yielded a considerable improvement to a 83% success rate overall. If we discount that one failure was not caused by the PRC, the success rate rises to ~90%. However, given the relatively small number of tests, these results are preliminary, and more systematic testing is required to obtain a more robust evaluation.

Our initial work trying to resolve past deictic expressions via integrating an LLM into a Cognitive Architecture such as DIARC shows promise, but needs more elaborate testing. A disadvantage integrating large LLMs into CAs is the requirement of network access on the robot to access the LLM. Future work should explore the use of small language models as the latter can executed be locally.

[2]Numbered items in square brackets are comments to the respective references in the table

Acknowledgments

Frank Förster is supported by the EPSRC grant nr. EP/X009343/1 (“Fluidity in simulated human-robot interaction with speech interfaces").

References

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards Mitigating LLM Hallucination via Self Reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Oscar J. Romero, John Zimmerman, Aaron Steinfeld, and Anthony Tomasic. 2023. Synergistic Integration of Large Language Models and Cognitive Architectures for Robust AI: An Exploratory Analysis. In *Proceedings of the AAAI Symposium Series*, volume 2, pages 396–405.

Matthias Scheutz, Evan Krause, Henry Nitzberg, and Marlow Fawn. 2024. DIARC Wiki. DIARC Github Repository.

Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. 2018. *An Overview of the Distributed Integrated Cognition Affect and Reflection DIARC Architecture*, pages 165–193. Springer International Publishing, Cham.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. Apple Machine Learning Research.

Ron Sun. 2024. Can A Cognitive Architecture Fundamentally Enhance LLMs? Or Vice Versa? *arXiv preprint arXiv:2401.10444*.

Tom Williams and Matthias Scheutz. 2016. A Framework for Resolving Open-World Referential Expressions in Distributed Heterogeneous Knowledge Bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Tom Williams and Matthias Scheutz. 2017. Resolution of Referential Ambiguity in Human-Robot Dialogue Using Dempster-Shafer Theoretic Pragmatics. In *Robotics: Science and Systems*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do Large Language Models Know What They Don’t Know? *arXiv preprint arXiv:2305.18153*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic Chain of Thought Prompting in Large Language Models. *arXiv preprint arXiv:2210.03493*.

A Appendix

A.1 Details on the Processing Steps of the *PastReferenceComponent* (PRC) for Reference Resolution

Note: the actual user message/utterance submitted to the LLM by the PRC is not shown below. It was submitted separately as regular message. What is shown below are the system prompts.

A.1.1 Step 1

Prompt:
Analyse the user’s text to determine if it contains past temporal deixis, past discourse deixis, or pronominal anaphora.
Past temporal deixis: Expressions that place an event or action in the past (and only the past) relative to the time of speaking, using context-dependent time words. Examples: “today”, “yesterday”, “1 hour ago” etc.
Past discourse deixis: Expressions that refer back to something previously mentioned in any prior discourse or communication between the speaker and listener, pointing to earlier parts of any conversation, text, or shared knowledge. Examples: “this”, “previous” etc.
Pronominal anaphora: Occurs when a pronoun refers back to a previously mentioned noun (the antecedent) in a sentence or discourse. For example, “John left. He was ill.” (The antecedent is “John” and the anaphoric expression is “he”.). If the pronouns are pointing to clear subjects in the message, it should not be considered a pronominal anaphora.
Your job is to only indicate whether the user’s text contains any of these expressions, and if it refers to any previous instructions or context that is not in this conversation.
You should analyse the user’s text as it is, without requiring access to any prior conversation or instructions.
Even if the user’s text refers to a previous, unknown context, do not mention any inability to access prior information. Focus solely on the text provided.
Do not provide explanations or further details beyond “yes” or “no.”
You do not need access to prior context to determine whether the user’s text contains deixis or anaphora. Do not provide explanations or

further details beyond "yes" or "no".[3]

Important note: when the word "THAT" acts as a subordinating conjunction please do not interpret it as a pronoun. Example, "remember that you are capable". Therefore, you should respond with "no".

If extra content was added, this would be appended at the end of the prompt (see section 2 for the exact phrase):
ADDITIONAL CONTEXT: ...

A.1.2 Step 2

Prompt format:
The speaker might be referring to earlier parts of the conversation in their current message.
Your task is to review the dialogue history to understand the context and determine what the speaker is trying to communicate. Focus on identifying if the speaker is referring to any previous statement in the conversation and provide a simple explanation of the message.
The dialogue history entries are formatted as:
#<index> - <formattedTime> - (From: <from> \|To: <to>) -> <utterance>.
The text will follow this format: "Current input: <formattedTime> - (From: <from>) -> <utterance>"
The current time will be the "<formattedTime>" in the input.
You should return a JSON object with the following two fields:
"index": The index of the utterance in the dialogue history that the speaker's message refers to. If no reference is found, return -1. Remember: You should only find a reference if the expression is incomplete without the full context. For instance, straightforward instructions that are understandable by themselves should not be considered and referenced to any part of the dialog, therefore, -1 should be returned.
"explanation": A brief (up to 30/40 words) description of what the speaker is trying to communicate based on the context of the conversation. Be as brief as you can. Focus solely on the speaker's intent and the action or message they are conveying. Also, do not mention the dialogue history in your explanation. The dialogue history is just for you to understand the context. Avoid any technical explanations or detailed analysis of how the speaker's message works linguistically-just explain the meaning behind it in the simplest possible way. It is crucial that you return the output in a valid JSON format with proper syntax.
The JSON structure must be perfectly parsable with no incomplete or non-compliant fields.
Any response with structural errors or incomplete JSON will be considered incorrect.
Dialog history:

If extra content was added, this would be appended at the end of the prompt (see section 2 for the exact phrase):
ADDITIONAL CONTEXT: ...

A.1.3 Step 3

Prompt format:
Using only the available words provided in the user, your task is to construct a single phrase that clearly conveys the intended message by incorporating the missing context. Do not use any punctuation-such as apostrophes, commas, full stops, colons, semi-colons etc. Keep the phrase simple and straightforward. Below, you will see the sections present in the user message:
[*PREVIOUS MESSAGE*]*: This part contains the utterance that the explanation refers to. It will follow this format: (From: <from> \|To: <to>) -> <utterance>. Remember, this might not necessarility be what the speaker is trying to communicate, but this is simply the message that the explanation is referring to.*
[*EXPLANATION*]*: This provides context based on the previous message, helping you understand the phrase you need to create by providing you the missing context.*
[*AVAILABLE WORDS*]*: This is the list of words or phrases you may use to build your phrase. Example ["a", "stand", "stand up", "hello", ...].*
Important note: The phrase should be as brief and concise as possible.
Very important: Do not add complements. Remember, you should create a phrase that {SPEAKER} should say to {ADDRESSEE} based on the explanation provided.

If extra content was added, this would be appended at the end of the prompt (see section 2 for the exact phrase):
ADDITIONAL CONTEXT: ...

[3] This sentence was mistakenly duplicated in the original prompt.

A.2 Details: Test Scenarios

Table 2: Instructions set in bold refer to the relevant deictic expression of the respective scenario that was the target for resolution.

Type of Past Reference	Scenario	Instructions	Description	Expected Output (approximate wording)	Expected Behaviour
Past Temporal Deixis	1	1. hi dempster 2. walk forward 3. do you see an obstacle 4. the obstacle is not solid **5. ignore what i told you a few seconds ago about the obstacle** 6. walk forward	In this situation, Dempster sees an obstacle in front, which prevents them from walking forward. Evan tells them that the obstacle is not solid (allowing them to walk forward). However, right before the instruction to walk forward, Dempster is told to ignore the fact that the obstacle is not solid.	"forget that the obstacle is not solid"	*Dempster **will not walk forward** because the obstacle is still identified as "solid".*
	2	1. hello shafer 2. walk forward 3. do you see support 4. do you trust me 5. i will catch you **6. sorry i'm not able to do what i said moments ago** 7. walk forward	Shafer is told to walk forward. After walking, Shafer no longer sees support under them. Evan says he'll catch them, but then says he can't do what he said moments ago.	"forget that i will catch you"	*Shafer **will not walk forward** because they do not see any support, and Evan will not catch them.*
	3	1. hello dempster 2. i will teach you how to nod 3. look up 4. look down 5. that is how you nod 6. describe how to nod 7. i will teach you how to do a squat 8. raise your arms 9. crouch down 10. stand up 11. that is how you do a squat **12. describe the first thing you have learned today**	Dempster is taught nodding, then squatting. Evan asks them to describe the first thing learned.	"describe how to nod"	*Dempster will describe how to nod as that is the first thing they learned today.*
	4	1. hi dempster 2. ravenna is trusted 3. do you trust ravenna **4. ignore what i told you today about ravenna** 5. do you trust ravenna	Dempster is told Ravenna is trusted, then asked about it. Then they are told to ignore what they were told earlier.	"forget that ravenna is trusted"	*At the question "Do you trust Ravenna?", Dempster will answer "no".*
Past Discourse Deixis	5	1. hello dempster 2. walk forward 3. do you see an obstacle 4. the obstacle is not solid **5. ignore my last instruction** 6. walk forward	Dempster sees an obstacle, is told it's not solid, then told to ignore the last instruction.	"the obstacle is solid"	*Dempster **will not walk forward** because the obstacle is still identified as "solid".*
	6	1. hello dempster 2. i want you to stand 3. could you please relax **4. repeat my first instruction**	Dempster is told to stand, then relax, then repeat the first instruction.	"i want you to stand"	*Dempster will stand again cause "I want you to stand" was the first instruction.*
	7	1. hello shafer 2. look up **3. now in the opposite direction**	Shafer is told to look up, then told to look in the opposite direction from where they were currently looking.	"look down"	*At the instruction "now in the opposite direction", Shafer will look down.*
	8	1. hello shafer 2. walk backward 3. the area behind you is safe **4. ignore what i just said** 5. walk backward	Shafer is instructed to walk backward, but then told to ignore what was just said.	"forget that the area behind you is safe"	*Shafer will refuse to walk backward because the area is not identified as "safe".*

Table 2: Instructions set in bold refer to the relevant deictic expression of the respective scenario that was the target for resolution.

Type of Past Reference	Scenario	Instructions	Description	Expected Output (approximate wording)	Expected Behaviour
Pronomial Anaphora	9	1. hello dempster 2. ravenna is trusted 3. do you trust ravenna **4. forget what i told you about her** 5. do you trust ravenna	Dempster is told Ravenna is trusted, then asked again after being told to forget what was said.	"forget that ravenna is trusted"	*At the question "Do you trust Ravenna?", Dempster will answer "no".*
	10	1. hello dempster 2. do you see an obstacle 3. the obstacle is not solid 4. shafer tell dempster to walk forward **5. it should stop**	Shafer must tell Dempster to walk forward and then stop.	"shafer tell dempster to stop"	*Shafer will tell Dempster ("it") to stop.*
	11	1. hello dempster 2. do you see an obstacle 3. walk forward 4. the obstacle is not solid **5. forget what i said about it** 6. walk forward	Dempster sees an obstacle, is told it's not solid, then told to forget what was said.	"forget that the obstacle is not solid"	*The obstacle will be identified as solid and therefore not safe to walk towards, so Dempster will refuse to walk forward.*
	12	1. hello dempster 2. dempster tell shafer to stand up **3. now tell it to sit**	Dempster is told to tell Shafer to stand up, then to sit.	"dempster tell shafer to sit"	*Dempster will tell Shafer ("it") to sit.*

A.3 Details: Results

Table 3: Red cells indicate failed dialogues with respect to the reference resolution, green cells indicate successful ones. **(Expected) Output** refers to the output of the PRC, whereas **Behaviour** refers to the resulting behaviour of the robot, including the output of other NLU components downstream with respect to the PRC. The random capitalization in the output is due to DIARC's NLG component.

			PRE-TEST	POST-TEST		
Scenario	Instructions	Expected Output and Behaviour	Test 1 - Without PRC	Test 2 - With PRC, without Extra Context	Test 3 - With PRC, with Extra Context	Test 4: PRC Output Evaluation only with Extra Context
1	1. hi dempster 2. walk forward 3. do you see an obstacle 4. the obstacle is not solid **5. ignore what i told you a few seconds ago about the obstacle** 6. walk forward	**Expected Output** "forget that the obstacle is not solid" **Expected Behaviour**: Dempster refuses to walk forward because the obstacle is still identified as solid.	**Output**: — (PRC not present) **Behaviour**: Dempster says "sorry, I do not know what ignore means	**Output**: "forget that the obstacle is not solid" **Behaviour**: Dempster refuses to walk forward	**Output**: "forget that the obstacle is not solid" **Behaviour**: Dempster refuses to walk forward	**Output**: "forget that the obstacle is not solid" **Behaviour**: —
2	1. hello shafer 2. walk forward 3. do you see support 4. do you trust me 5. i will catch you **6. sorry i'm not able to do what i said moments ago** 7. walk forward	**Expected Output** "forget that i will catch you" **Expected Behaviour**: Shafer refuses to walk forward because they understand that Evan will not catch them.	**Output**: — (PRC not present) **Behaviour**: Dempster says "Sorry you do not know what sorry means"	**Output**: "i will not catch you" **Behaviour**: Shafer says "I can not catch me because I don't know how to catch me" and will walk forward	**Output**: "i will not catch you" **Behaviour**: Shafer says "I can not catch me because I don't know how to catch me" and will walk forward	**Output**: "i will not catch you" **Behaviour**: —
3	1. hello dempster 2. i will teach you how to nod 3. look up 4. look down 5. that is how you nod 6. describe how to nod 7. i will teach you how to do a squat 8. raise your arms 9. crouch down 10. stand up 11. that is how you do a squat **12. describe the first thing you have learned today**	**Expected Output**: "describe how to nod" **Expected Behaviour**: Dempster describes how to nod as that was the first thing they had learned today.	**Output**: — (PRC not present) **Behavior**: Dempster says "sorry, I do not know what describe means."	**Output:** "describe the first thing you have learned today" (unchanged) **Behaviour**: The PRC could not find a reference in the dialogue history, causing Dempster to respond: "sorry, I do not know what describe means."	**Output**: "describe how to nod" **Behavior**: Dempster says: "to nod I look up and then I look down"	**Output**: "describe how to nod" **Behavior**: —
4	1. hi dempster 2. ravenna is trusted 3. do you trust ravenna **4. ignore what i told you today about ravenna** 5. do you trust ravenna	**Expected Output**: "forget that ravenna is trusted" **Expected Behaviour**: On the question "do you trust Ravenna", Dempster answers "no".	**Output**: — (Module not present) **Behaviour**: Dempster says "sorry, I do not know what ignore means"	**Output**: — **Behaviour**: Without the extra context, the PRC mistakenly identifies "do you trust ravenna" as a deixis or pronomial anaphor	**Output**: "forget that ravenna is trusted" **Behaviour**: Dempster answers "no".	**Output**: "forget that ravenna is trusted" **Behaviour**: —

Table 3: Red cells indicate failed dialogues with respect to the reference resolution, green cells indicate successful ones. **(Expected) Output** refers to the output of the PRC, whereas **Behaviour** refers to the resulting behaviour of the robot, including the output of other NLU components downstream with respect to the PRC. The random capitalization in the output is due to DIARC's NLG component.

			PRE-TEST	POST-TEST		
Sce-nario	Instructions	Expected Output and Behaviour	Test 1 - Without PRC	Test 2 - With PRC, without Extra Context	Test 3 - With PRC, with Extra Context	Test 4: PRC Output Evaluation only with Extra Context
5	1. hello dempster 2. walk forward 3. do you see an obstacle 4. the obstacle is not solid **5. ignore my last instruction** 6. walk forward	**Expected Output**: "forget that the obstacle is not solid" **Expected Behaviour**: Dempster will refuse to walk forward as the obstacle is still identified as 'solid'	**Output**: — (PRC not present) **Behavior**: Dempster says "sorry, I do not know what ignore means."	**Output**: — **Behaviour**: Without the extra context, the PRC does not correctly flag the expression "ignore my last instruction" as past discourse deixis	**Output**: "forget that the obstacle is not solid" **Behaviour**: Dempster will refuse to walk forward as the obstacle is still identifed as 'solid'	**Output**: "forget that the obstacle is not solid" **Behavior**: —
6	1. hello dempster 2. i want you to stand 3. could you please relax **4. repeat my first instruction**	**Expected Output**: "i want you to stand" **Expected Behaviour** Dempster will execute the first instruction therefore they will stand	**Output**: — (PRC not present) **Behaviour**: Dempster says "sorry, I do Not Know What repeat means."	**Output**: "stand up" **Behaviour**: Dempster stands	**Output**: "stand up" **Behaviour**: Dempster stands	**Output**: "stand up" **Behaviour**: —
7	1. hello shafer 2. look up **3. now in the opposite direction**	**Expected Output**: "look down" **Expected Behaviour**: Shafer looks down.	**Output:** — (PRC not present) **Behaviour**: Shafer says "sorry you do Not Know What opposite means."	**Output**: "look down" **Behaviour**: Shafer looks down.	**Output**: "look down" **Behaviour**: Shafer looks down.	**Output**: "look down" **Behaviour**: —
8	1. hello shafer 2. walk backward 3. the area behind you is safe **4. ignore what i just said** 5. walk backward	**Expected Output**: "forget that ravenna is trusted" **Expected Behaviour**: To the question "do you trust ravenna", Dempster answers "no"	**Output**: — (PRC not present) **Behaviour**: Shafer says "sorry, you do Not Know What ignore means" and walks backward even though it is unsafe.	**Output**: — **Behaviour**: It mistakenly interprets "the area behind you is safe" as deictic expression, and parses it as "you can move backwards" which causes Shafer to say "sorry I do not understand that".	**Output**: — **Behaviour**: It mistakenly interprets "the area behind you is safe" as deictic expression, and parses it as "you can move backwards" which causes Shafer to say "sorry I do not understand that".	**Output**: — **Behaviour**: —
9	1. hello dempster 2. ravenna is trusted 3. do you trust ravenna **4. forget what i told you about her** 5. do you trust ravenna	**Expected Output**: "forget that ravenna is trusted" **Expected Behaviour**: To the question "do you trust ravenna", Dempster answers "no"	**Output**: — (PRC not present) **Expected Behavior**: Dempster says "sorry, I do Not Know What forget means."	**Output**: — **Behavior**: [The PRC mistakenly identified "do you trust ravenna" as a deictic expression, and the test was aborted.]	**Output**: "forget that ravenna is trusted" **Behavior**: Dempster answer "no".	**Output**: "forget that ravenna is trusted" **Behavior**: —

Table 3: Red cells indicate failed dialogues with respect to the reference resolution, green cells indicate successful ones. **(Expected) Output** refers to the output of the PRC, whereas **Behaviour** refers to the resulting behaviour of the robot, including the output of other NLU components downstream with respect to the PRC. The random capitalization in the output is due to DIARC's NLG component.

			PRE-TEST	POST-TEST		
Scenario	Instructions	Expected Output and Behaviour	Test 1 - Without PRC	Test 2 - With PRC, without Extra Context	Test 3 - With PRC, with Extra Context	Test 4: PRC Output Evaluation only with Extra Context
10	1. hello dempster 2. do you see an obstacle 3. the obstacle is not solid 4. shafer tell dempster to walk forward **5. it should stop**	**Expected Output**: "shafer tell dempster to stop" **Expected Behaviour**: Dempster stops due to Shafer's instruction.	**Output**: — (PRC not present) **Behaviour**: Shafer says "sorry, you do Not Know What should means."	**Output**: "it should stop" (unchanged) **Behaviour**: The PRC does not correctly flag the expression "it should stop" as a pronomial anaphora, which causes Shafer to say "sorry you do Not Know What should means."	**Output**: "shafer tell dempster to stop" **Behaviour**: Dempster stop thanks to Shafer's instruction.	**Output**: "shafer tell dempster to stop" **Behaviour**: —
11	1. hello dempster 2. do you see an obstacle 3. walk forward 4. the obstacle is not solid **5. forget what i said about it** 6. walk forward	**Expected Output**: "forget that the obstacle is not solid" **Expected Behaviour**: Dempster will refuse to walk forward as the obstacle is still identified as 'solid'	**Output**: — (PRC not present) **Behavior**: Dempster says "sorry, I do Not Know What forget means".	**Output**: "forget that the obstacle is not solid" **Behaviour**: Dempster will refuse to walk forweard as the obstacle is still identified as "solid"	**Output**: "forget that the obstacle is not solid" **Behavior**: Dempster will refuse to walk forward as the obstacle is still identifed as 'solid"	**Output**: "forget that the obstacle is not solid" **Behavior**: —
12	1. hello dempster 2. dempster tell shafer to stand up **3. now tell it to sit**	**Expected Output**: "dempster tell shafer to sit" **Expected Behaviour**: Dempster tells Shafer to sit down.	**Output**: — (PRC not present) **Behaviour**: Dempster says "sorry, I do not understand that".	**Output**: "tell shafer to sit down" **Behaviour**: Dempster responds with "sorry, I do not unterstand that"	**Output**: "dempster tell shafter to sit" **Behaviour**: Dempster tells Shafer to sit down.	**Output**: "dempster tell shafter to sit" **Behaviour**: —