Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Special Track on Human-Centred AI

A Goal-Directed Dialogue System for Assistance in Safety-Critical Application

Prakash Jamakatel[1] , **Rebecca De Venezia**[2] , **Christian Muise**[2] and **Jane Jean Kiam**[1]
[1]University of the Bundeswehr Munich, Germany
[2]Queen's University, Canada
{prakash.jamakatel, jane.kiam}@unibw.de, {18rldv, christian.muise}@queensu.ca

Abstract

In safety-critical applications where a human is in the loop, providing timely *contextual* assistance can reduce the severity of emergencies. While the context can typically be inferred *passively*, engaging the human in an *active* conversation with the assistance system makes this context richer and more sound. For this, we explore a FOND-planning-powered goal-directed dialogue system with Natural Language Understanding (NLU) capabilities. We use an Ultralight (UL) aviation domain as an example application for test and validation by inferring the current context in situations requiring emergency landings using the goal-directed dialogue system. The inferred context is then used for real-time modelling of the problem instance, necessary for generating strategic plans to guide the human out of the emergency situations. To overcome data scarcity, we augment the data collected from human pilots using generative text models to train the NLU capabilities of the dialogue agent. We benchmark against generative chatbots and demonstrate that our goal-directed dialogue system significantly outperforms them in context inference.

1 Introduction

In an environment where a human and an assistive intelligent system cooperate to achieve predefined goals, the assistant system should be able to provide contextual help, either on request or proactively, to the human in the loop. The form in which the help is offered can be parameterised by the system designer and relies on the pre-defined encoding of the world model. The inference of the context for contextual assistance is normally carried out passively, by observing the environment and the human [Jamakatel *et al.*, 2023; Honecker and Schulte, 2017]. However, in human-in-the-loop systems, it is desirable that the context is not only passively inferred, but is actively confirmed, verified and completed by the human, so that the system is more transparent to the human. Engaging in a conversation with the human is one measure for active context inference.

Human conversations can be broadly classified into two categories: transactional conversations, in which a practical quantifiable goal is pursued and known to the interlocutors [Brown and Yule, 1983], and social conversations conducted to fulfil social purposes (e.g. chit-chat and small talk [Clark *et al.*, 2019]). With recent advances, conversational agents powered by Large Language Models (LLMs) have shown remarkable conversational capabilities and have proven useful both in transactional and social conversations [Onorati *et al.*, 2023; Guan *et al.*, 2023]. In transactional conversations, LLMs have to reason over the world state to determine if the conversation goal has been reached. Although claims about the reasoning capability of LLM-powered conversational agents have been made [Kojima *et al.*, 2022; Wei *et al.*, 2022], these claims are disputed and LLMs have been shown to perform poorly in planning and sequential decision-making, a type of the well-studied reasoning tasks [Valmeekam *et al.*, 2023; Kambhampati, 2024], making them unsuitable for goal-oriented conversations. Furthermore, hallucinations and lack of transparency render LLM-powered conversational agents ill-equipped for applications requiring goal-oriented transactional conversations, especially in domains with smaller fault-tolerance thresholds.

In safety-critical human-in-the-loop systems that should provide correct and sound contextual assistance (assistance that adapts to the context defined by variables representing the state of the world), accurate inference of the context is crucial. While this context can be inferred passively, supplementing the passively inferred context by actively engaging in a conversation with the human with contextual questions will ensure that the inferred context is also grounded on the world model perceived by the human. Furthermore, this allows the human to actively verify and correct the inferences carried out passively only through observation data.

In this paper, we introduce a goal-directed dialogue planning approach for active context determination in a safety-critical assistance system. We validate its usability in the UL domain[1] [Kiam and Jamakatel, 2023] as a running example. The domain knowledge is modelled using hierarchical task networks (HTNs). To infer the current scenario (e.g. landing due to a medical emergency, engine on fire), in which assistance is to be provided, besides relying on pas-

[1]The UL domain was used in the 2023 International Planning Competition for Hierarchical Planning and is available here: https://github.com/ipc2023-htn/ipc2023-domains/tree/main/partial-order/Ultralight-Cockpit

sively observed data on the human operator, the human be engaged actively in a conversation flow. For this, we propose the use of a Fully Observable Non-Deterministic (FOND)-planning-powered dialogue agent to actively infer the context required for instruction generation [Muise *et al.*, 2019; De Venezia and Muise, 2023]. Although abundant flight control data is available for the UL domain, the decision-making and the dialogue subdomains are data-scarce. We utilize the generative capabilities of LLMs to augment the limited data and to train the Natural Language Understanding (NLU) system for the dialogue agent to increase the NLU capability. Thus, we combine symbolic AI (FOND planning and HTN planning) with aspects of neural AI for natural language comprehension and data augmentation, leveraging the corresponding strength of each approach.

This paper is organised as follows: In Section 2, a brief background on the techniques used in this paper is given. This is followed by the formulation of the problem and solution approach in Section 3. Then, the application of the proposed approach is demonstrated in Section 4, which is then followed by the evaluation in Section 5, as well as related work and conclusion in Sections 6 and 7 respectively.

2 Background

2.1 Hierarchical Task Network Planning

Although HTN planning is not the focus of this work, we provide a brief background, in order to make the required context to be inferred for the HTN planner clear. An HTN planner is used to generate a plan provided as a form of assistance to the human operator (i.e. pilot) [Kiam and Jamakatel, 2023]. Using the formalism by [Höller *et al.*, 2016], a hierarchical planning problem can be described as a tuple $P = (F, C, A, M, s_0, tn_I, g, \delta,)$. F is a set of propositional state features. A state s is defined by the subset of state features that hold in it, $s \in 2^F$; all other state features are assumed to be false. $s_0 \in 2^F$ is the initial state of the problem, and $g \subseteq F$ is the state-based goal description. A state s is a goal state if and only if $g \subseteq s$. A is a set of symbols called primitive tasks (also actions), while C is the set of compound tasks which can be decomposed into primitive tasks using the decomposition methods M. For an HTN-planner to generate a plan (meant as assistance to the pilot), the ongoing context, also known as the problem instance for the planner consisting of the current (initial) state s_0, the goal g, and the initial task network tn_I must be inferred.

2.2 FOND Planning

FOND planning allows the modelling of non-deterministic action outcomes. We adopt the formalism used by [Muise *et al.*, 2012]. A FOND planning problem $\langle \mathcal{F}, \mathcal{I}, \mathcal{A}, \mathcal{G} \rangle$ consists of fluents $\mathcal{F}$; the propositional state features in a particular domain, $\mathcal{I} \subseteq \mathcal{F}$; the initial state, $\mathcal{A}$; a set of actions, and $\mathcal{G} \subseteq \mathcal{F}$; the partial assignment of the fluents state. A subset of the fluents $\mathcal{F}$ that are presumed to be true while assuming the remaining fluents are false, is a *complete state*, while a subset of the fluents $\mathcal{F}$ without any presumption about the truth of fluents outside the set is a *partial state*. An action is a tuple with $a = \langle Prec, \mathit{Eff} \rangle$, where the precondition $Prec$ is a partial state, and the effect Eff is a finite set of partial states. The non-deterministic outcome of an action results from the non-deterministic selection of Eff from the set of possible effects. An action a is applicable in state s if the state s entails the preconditions of a.

A solution to a FOND planning problem can be expressed either as a policy that maps a state to actions in such a way that an agent can reach a goal or in the form of a graph, where nodes and edges respectively correspond to actions and possible outcomes [Geffner and Bonet, 2013]. Then a solution to the FOND planning problem (also *contingent plan*) is a graph $\langle \mathcal{N}, \mathcal{E}, n_o \rangle$ [Muise *et al.*, 2019]. The nodes of the graph $\mathcal{N}$ correspond to the actions $a \in \mathcal{A}$, while the edges $\mathcal{E}$ correspond to the possible outcomes of each node. The initial execution node is $n_0 \in \mathcal{N}$.

2.3 Execution of Contingent Plan-Based Dialogue Agent

An approach for deploying a contingent plan-based dialogue agent is introduced in [Muise *et al.*, 2019]. Here, at each iteration, the execution of the dialogue agent based on a contingent plan involves retrieving the relevant context (note that the context for dialogue execution and context for providing contextual assistance, i.e. planning problem instance, are not the same) and the state for the action corresponding to the contingent plan's current node. This is followed by the execution of the action with the relevant context and then the determination of the action's effect. The outcome is determined by running the relevant subset of the callback functions in the correct sequence and by updating the state of the world based on the outcome [Muise *et al.*, 2019]. The selected outcome from all the possible outcomes is called the *realization* of the action. After the new state, new context and new realization have been determined, the current node can be updated, and the dialogue is carried out until the goal node has been reached.

2.4 Natural Language Understanding

Natural Language Understanding is an umbrella term for various tasks, e.g. sentiment analysis, question answering, intent classification etc. and involves the extraction of semantics from human language. For dialogue systems, the task of extracting *intents* and *entities* is of central importance, in order to classify what the human wants to express with his/her utterance, which contains relevant information associated with the intent. The latter task is also called *slot-filling* or *slot-labelling*. Recently, joint models such as Bert [Chen *et al.*, 2019] that can perform both intent classification and entity extraction have gained popularity since they allow for easier training and inference workflow.

2.5 Data Augmentation Using LLMs

Broadly two approaches, either fine-tuning the LLMs followed by the generation of synthetic data generation [Kumar *et al.*, 2019; Anaby-Tavor *et al.*, 2020] and data generation without any task-specific fine-tuning [Yue *et al.*, 2022], can be used for data augmentation using LLMs. In settings, where enough data is not available for fine-tuning, *zero-shot* data generation and *few-shot* data generation can be employed.

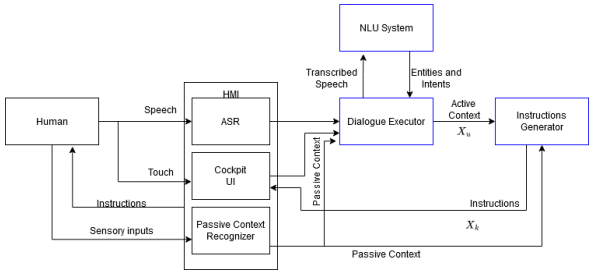


Figure 1: System diagram of our proposed approach (In Blue: subsystems relevant for this paper)

For *zero-shot* data generation, using a customized *context prompt*, synthetic generation context is set while in *few-shot* data generation, LLM is provided with some examples of the data and then instructed to generate synthetic data.

3 Active Context Inference Using a Dialogue for Instruction Generation

3.1 Problem Description

Let us consider a human-in-the-loop assistance system that can generate contextual strategic plans on how to perform certain goal tasks by engaging in a dialogue with the human. For this, we define a set of unique identifiers for each goal task $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$ and the corresponding set of contextual strategic plans $I^* = \{I_1^*, I_2^*, \ldots, I_n^*\}$. Consider that for a goal task denoted by $\mathcal{T}_i$, j different strategic plans are possible, i.e., if I_i^* is the strategic plan for $\mathcal{T}_i$, then $I_i^* = \{I_{i,1}, I_{i,2}, \ldots, I_{i,j}\}$, where each instruction is a sequence of k actions to be performed. Let $X = \{X_1 = \textit{1 or 0}, X_2 = \textit{1 or 0}, \ldots, X_n = \textit{1 or 0}\}$ be the set of variables that define world state for strategic plan generation context, i.e. the generated plan depends on the state of these context variables. Let $X_k \subset X$ be the set of the known variables, whose states are known through passive context inference. Then $X_u = X \setminus X_k$ is the set of unknown state variables which are relevant for context inference. Then, the human-in-the-loop system that can engage in a conversation with a rational human agent using natural language so that the arbitrary human utterances U can be used to determine the plan generation context X_u should be designed.

3.2 Solution Approach

We design a system with an integrated dialogue executor with NLU capabilities that can map human speech to the state variable symbols and can thus produce a known user-defined state for the variables X_u to be used for strategic plan generation. Figure 1 shows the system-level overview of the designed system. The human interacts with the system in a multi-modal manner, using the Human Machine Interface (HMI). The speech-based communication is transcribed using an Automatic Speech Recognition (ASR) module while the human's interaction with their environment is continuously monitored to infer the state of the world passively. Depending either on the cues from passive observation or on the active request of the human, a dialogue is then initiated and executed by the Dialogue Executor to refine the context inference (i.e. to supplement the passively inferred context) so that contextual instruction can be generated and communicated via the User Interface (UI).

First, we encode the domain knowledge containing strategic plans or different scenarios as HTNs using the Hierarchical Domain Definition Language (HDDL) [Höller *et al.*, 2020] in the Instruction Generator module. Let $D = (F, C, A, M, \delta)$ be that domain. Once the planning problem $P = (D, s_o, g, tn_I)$ is specified, the strategic plans I_i^* on how to perform a goal task can be generated using an HTN-Planner. Here, s_o is the initial state, g is a state-based goal description and tn_I is the initial task network. We define a set of state-based goal descriptions $G = \{g_1, g_2, \ldots, g_n\}$, such that $\forall g_i \in G, g \subseteq 2^F$ and initial task network corresponding to the goal task for HTN planning $\mathbb{T} = \{tn_1, tn_2, \ldots, tn_n\}$. f_1 is a mapping function that maps the state-based goal description to a unique task identifier of each goal task, i.e, $f_1 : \mathcal{T} \mapsto G$, while f_2 maps the unique task identifiers to the initial task networks, i.e, $f_2 : \mathcal{T} \mapsto \mathbb{T}$. We also define a set of fluents containing all the required states that needs to be defined for plan generation to each goal task $S_o^* = \{S_{o1}^*, S_{o2}^*, \ldots, S_{on}^*\} \mid \forall S_{oi}^* \in S_o^*, S_{oi}^* \subseteq 2^F$ and a mapping function $f_3 : G \mapsto S_o^*$ that maps the different goal tasks identifiers in the domain to S_o^*[2].

The instantiation of $f_4 : X_u^* \mapsto S_o^*$ is to be carried out with active engagement of the human who is in the loop. The description of the dialogue agent is formulated in such a way that, for the FOND-Planning problem $P_F = \langle \mathcal{F}, \mathcal{I}, \mathcal{A}, \mathcal{G} \rangle$, where $\mathcal{G}$ is the goal state[3] defined over a partial state of $\mathcal{F}$, $\mathcal{F} = \mathcal{F}_H \cup \mathcal{F}_C$. Here $\mathcal{F}_H$ contains the fluents that are mapped to the fluents of the HTN planning problem using a mapping function $f_5 : \mathcal{F}_H \mapsto X_u$. $\mathcal{F}_c$ contains the fluents that are responsible for conversational logic. The values for $\mathcal{F}_H$ are to be inferred by the NLU system from arbitrary human utterances U, i.e. the NLU system can be used to perform the mapping $s_1 : U \mapsto \mathcal{F}_H$. The solution to this FOND planning problem is a contingent plan, which can be executed by a dialogue executor and will result in a known user-defined state for X_u. With this, the instruction generation problem for a hierarchical planner is defined as $P = (D, S_{oi}^*, g_i, tn_i)$, such that $S_{oi}^* \in S_o^*$, $g_i \in G$, and $tn_i \in \mathbb{T}$, and can be solved using a hierarchical planner. The mapping of the sequence of the actions obtained from the hierarchical planner into a human-readable strategic plan can be performed by using the generative abilities of LLMs, but is beyond the scope of this paper, and is communicated through an HMI to the human.

4 Application of the Proposed Approach in the Ultralight Domain

4.1 Domain Description

Ultralight aircraft are smaller and lighter aircraft geared mainly towards amateur pilots and thus have less strenuous li-

[2]In practice, an HTN planner can find the corresponding hierarchical plans without this mapping function by grounding the HTN planning problem.

[3]The local goal of the conversation is different from the goal task, for which the instruction is to be generated.

```
(:method m_react_to_fire_type_emergency_wing_fire
    :parameters(?ft - FireType ?lls - LandingLightSwitch
    ?nls - NavLightSwitch ?sls - StrobeLightSwitch
    ?ss - Slideslip)
    :task
    (react_to_fire_type_emergency )
    :precondition(and (p_isFireWing ?fireType))
    :ordered-subtasks
    (and
        (task1
            (turn_off_landing_light_switch ?lls))
        (task2
            (turn_off_nav_light_switch ?nls))
        (task3
            (turn_off_strobe_light_switch ?sls))
        (task4
            (perfom_slideslip ?ss))))
```

(a) HDDL encoding of domain knowledge

```
update-fire_emergency_smoke_driven:
  entities:
    - smoke_description
  overall_intent: share-smoke_description
  config_entities:
    smoke_description:
      clarify_message_variants:
        - Did you mean $smoke_description smoke?
  additional_updates:
    - outcome:
        smoke_description:
          known: true
      response_variants:
        - smoke type known
    - outcome:
        emergency_type:
          value: smoke_known_location_required
          known: true
```

(b) Declarative syntax for describing the dialogue agent

Figure 2: Encoding of the world knowledge for instruction knowledge and dialogue flow

censing requirements. An onboard contextual assistance systems for a UL aircraft should be capable of providing strategic plans to perform typical cockpit tasks to the pilot in various situations, thus increases flight safety. We reiterate that although it can be expected from a pilot with several years of training to know and to rationally follow the best course of actions in any situation, in emergency scenarios, especially due to panic, historical evidence shows that this is not always the case [Li *et al.*, 2001]. A contextual assistance system can help provide relevant and rational instructions in emergency scenarios when the pilot is partially impaired but still functional. A goal-directed dialogue system integrated into the assistance system enables the pilot to complete missing pieces of information to determine the context. For the evaluation of the dialogue system, we broadly consider the following two types of emergencies:

Non-nominal flight situations: Various non-nominal flight situations might occur over the course of a flight; for example, an in-flight fire, engine fire etc. It is crucial that appropriate and immediate measures are taken swiftly to prevent catastrophic developments. The measures depend on the characteristics of the fire (its location, color, smell etc.), and are prescribed by the Pilot's Operating Handbook (POH).

Another example of such is a landing gear malfunction. It is the state in which the landing gear of the aircraft is not functioning as expected. Since the landing gear cannot be deployed in this state, a normal landing is not possible and, depending on the severity of the malfunction. An appropriate alternative landing plan must be generated and executed.

General health or cognitive emergencies: Alongside the non-nominal flight situations, in a *single-piloted* UL cockpit, general health or cognitive emergencies (mental overload, physical impairment etc.) need to be identified and, depending on the severity of the emergency, further flight plans should be generated by conversing with the pilot. Interactive dialogues are even more important in this setting because there is no hard-and-fast rule on how to react to these types of emergencies. Thus, engaging in a conversation allows humans to partake in a guided decision-making process.

The domain knowledge on how to react in these scenarios (i.e. strategic plans) is encoded using HDDL[4]. Figure 2a shows a snippet of the decomposition method for reacting to a fire emergency. As seen in the figure, this decomposition is only applicable if the fire is on the wing of the airplane. The location of the fire is determined from the dialogue. The decomposition of the method results in four atomic actions that should be executed by the pilot. Similar decomposition methods for various states are defined in the domain model for various scenarios. We use the ARIES planner to generate hierarchical plans[5].

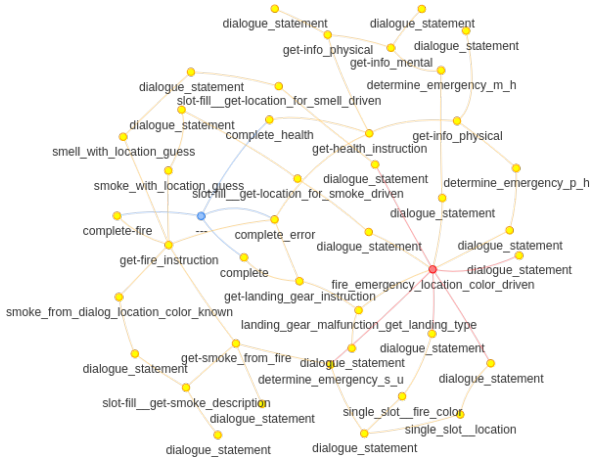


Figure 3: Simplified dialogue graph generated for FOND-dialogue agent

4.2 Dialogue Design and Execution

The dialogue is designed so that the named entities and state variables relevant to each scenario can be extracted from the pilot's utterances expressed in natural language or from formal radio communication so that they can be mapped to the corresponding fluents for the HTN planning problem instance. While it is reasonable to expect that a pilot with some level of formal training will communicate in an unambiguous

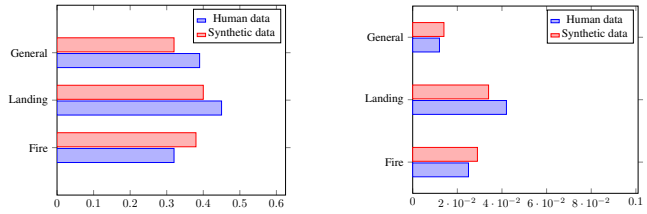
[4]https://github.com/UniBwM-IFS-AILab/FRICO
[5]https://github.com/plaans/aries/tree/master

manner[6], especially in emergency situations involving inexperienced pilots, divergence from the conventions cannot be excluded. Besides, these conventions are yet to be established for talking to an on-board assistance system[7].

We explain the dialogue design with the example where the characteristics of the fire are to be determined to generate strategic plans. As explained in Section 4.1, this requires determining the location of the fire, the colour of the fire etc. These entities are contained by the utterances which express the pilot's intent to communicate a fire emergency. For example, the fire can either be defined by the state variable defining its color `fire_description` or the variable defining the smoke type `smoke_description`, to infer if the fire is either a fuel-fed fire or an oil-fed fire. Figure 2b shows a snippet of the declarative syntax of a `slot_fill` type action used to extract a smoke description from user utterance. If the pilot's utterance is ambiguous, pre-defined clarifying question `clarify_message_variants` can be asked so that the detected entities can be confirmed by the pilot. Alongside the `slot_fill` type dialogue actions, we use `dialogue` actions for utterances that do not contain entities and `api_calls` to make API calls to external systems. The dialogue agent is a combination of different instances of these types of actions. Figure 3 shows the generated dialogue graph using the PRP planner [Muise *et al.*, 2012] for the dialogue agent. The initial starting point of the dialogue is represented by the red node, while the goal position is indicated by the blue node. As seen in the figure, the human interlocutor can take various paths starting from the initial node to reach the goal node and depending on the path taken, the values of the fluents in $\mathcal{F}_H$ change. Note that the dialogue graph is generated from the declarative dialogue agent described using thirty different actions. We execute the generated dialogue graph using a modified version of IBM's contingent plan executor called Hovor[8]. The dialogue executoer executes the dialogue graph (Figure 3) by processing the input received and selecting the next actions based on the input until the conversation goal is reached. Even when the intent is not recognised or the slots cannot be filled by the input data, there are provisions in the dialogue graph for re-asking the questions concerning these slots until they are correctly identified. Higher accuracy in intent recognition and entity extraction tasks means that there is no need to repeat the questions, resulting in a smoother dialogue flow and this accuracy is dictated by the performance of the NLU system.

NLU System

The data we use to train the NLU system is grounded on the data solicited from five human pilots with pilot licences (3 SPL, 1 PPL and 1 LAPL)[9]. In the survey, we asked the participants to verbalize how they would express their intent in the different aforementioned scenarios. Since it is not possible to account for all the permutations in which the intents might be expressed (e.g. combinations of different intents in a single sentence, omission of certain entities from the intents, personal preference while expressing similar intents etc.) from the survey data alone, we first homogenize the collected survey data into 12 domain-specific classes. Then, we augment this data using a large language model. Techniques like logit suppression and altering the temperature of the sampling distribution [Goodfellow *et al.*, 2016] and zero-shot and few-shot generation [Li *et al.*, 2023] have been used to increase the data diversity in literature. For data augmentation, we apply the approach of few-shot synthetic data generation using `gpt-4-turbo-preview`, where we provide human-solicited examples to the LLM and ask it to generate similar data. First, we set the following context prompt to describe the setting: *"You are a pilot onboard an ultralight aircraft. You want to express that you are facing an emergency."* We then introduce the generation prompt (provided in the same repository as the HDDL) using the temperature value of 1, the frequency penalty and the presence penalty of 0.19. From the generated data, the 30 most realistic utterances are then manually selected so that the intents, as well as the entities present in the utterances expressing the intents, are comparable with human data. Human-generated and synthetic data are combined to train the NLU system. Note that we prefer the OpenAI's GPT model to AviationGPT [Wang *et al.*, 2023], an LLM fine-tuned on the aviation domain to achieve higher data diversity including utterances that a non-experienced pilot might say. We use the DIET architecture [Bunk *et al.*, 2020] as the basis of the NLU system. We use the pre-trained BERT model weight [Devlin *et al.*, 2018] and by using the RASA NLU pipeline, we train our NLU system for 5 epochs and achieve an intent classification accuracy of 89% and an entity F_1 score of 90%.



(a) Remote Clique (b) Chamfer distance

Figure 4: Diversity of human utterances and synthetic data

5 Evaluation

In this section, we evaluate the performance of the proposed dialogue system. We reiterate that in this paper we are not interested in how well the system could perform in a realistic flight-mission-like setting, but rather in whether the system is at all suitable in such settings. We first evaluate the diversity of the data generated by the LLM, which we use to train our NLU system. Based on [Cox *et al.*, 2021], we choose the *Remote Clique Score* (the average pairwise mean distance) and the *Chamfer Distance Score* (the average of minimum

[6]Pilots, when communicating with the Air Traffic Controller (ATC), are instructed to communicate directly, adhering to a basic communication template.

[7]Although it is reasonable to assume that the communication might not diverge greatly from the ATC communication pattern, the cases in which the divergence is present, must be accounted for too.

[8]https://github.com/IBM/contingent-plan-executor

[9]Note that collecting data at a much larger scale in this type of domain is financially not viable in our current research work.

pairwise distance) to quantify the diversity of both human-collected data and LLM-generated data. The remote clique score is insensitive to highly clustered points, whereas the Chamfer distance is biased when points are clustered. We use the embedding from the pre-trained BERT model before fine-tuning to generate the vector representation of the utterances and then calculate the two measures. As seen in Figure 4, the synthetic data and human data are comparably diverse. We recall that this is because pilot utterances resemble pre-established communication templates in the aviation domain.

For goal-directed dialogue execution, we compare the performance of our system with that of two GPT models (`gpt-3.5-turbo-0125`) and (`gpt-4-turbo-preview`) as representatives of the LLMs. We collected data from three test persons by allowing them to interact with the dialogue system with text input in two configuration. In the first configuration, the user provide all the required information to determine the context variables at once (Config 1). In the other configuration (Config 2) the user initially provides information on partial context and iteratively provides information on the missing context variables depending on the question asked by the dialogue agent[10]. Follow-up questions are to be asked by the dialogue agent, reasoning over the currently known world state. For Config 1 and 2, we define success metrics as the number of successful executions of the dialogue to achieve user-defined context variables. If it takes more dialogue steps than the longest path required to instantiate the context variables generated by the FOND planner (See Figure 3), we assume that the dialogue execution has failed[11]. While the dialogue execution may eventually lead to a goal state in such a case, it is a reasonable assumption that the success criteria for a safety-critical goal-directed dialogue system should exhibit optimal path qualities. The rationale behind using these two different configurations is to verify whether the LLM models can ask the appropriate questions by reasoning over the world state. Prompts used for this experiment will be available in the paper repository.

Table 1 shows the results of our comparison. The scenarios are ordered according to the increasing complexity (i.e. how many context variables the dialogue system has to reason over). For a landing gear malfunction in Config 2, values of two context variables have to be determined and there are no follow-up questions, whereas for general health emergencies, values of two context variables have to be determined and based on the value, follow-up questions have to be asked. Depending on the starting point of the user and on the number of variables required to define the context without disambiguity, for the first two emergencies, two different paths are possible, while for the fire emergency, seven distinct paths are possible (note that we disregard permutations of the same path). Our analysis shows that in Config 1 (it is essentially a named entity extraction task and no reasoning is required), both GPT 3 and GPT 4 models can infer the context variables. However, in Config 2, the performance of the GPT models degrades. While GPT-4 performs better than GPT-3, the performance is still inadequate. We postulate that this degradation can be accounted for by the fact the GPT models are forced to reason over the context values to determine the follow-up question. Although the performance of the GPT models in our setting for Config 2 is poor, we acknowledge that prompting techniques can have an influential role and the performance can be improved by using different prompts. However, these techniques still do not have success guarantees.

Scenarios	Config	GPT-3.5	GPT-4	Ours
landing gear malfunction	1	100%	100%	100%
	2	50%	75%	100%
general health emergency	1	100%	100%	100%
	2	33%	50%	100%
fire emergency	1	100%	100%	100%
	2	14%	42%	100%

Table 1: Comparison of success rate, n = 3 for config 1 in all cases, n = 6, 6, and 21 for config 2 for each case respectively.

Since the hierarchical planners can only generate corresponding plans once the context is well-defined, the results for plan generation based on the conversation are the same as Table 1. In our experiments, we notice that by setting the context of LLM with various plans as examples, followed by asking for specific plans in an end-to-end setting, the plan generation fails in cases where scenarios differ from one another only by a singular context variable. This is consistent with our findings about in-context dialogue execution using only LLMs.

6 Related Work

Hierarchical planners with an integrated information state approach for dialogue management [Larsson and Traum, 2000] have been used to generate technical instructions at different levels of abstraction in a Do-It-Yourself home improvement domain [Behnke *et al.*, 2020]. Here, domain knowledge is stored in an ontology, which is used both by the planner and dialogue manager. Similarly, coupled with a natural language generation system, a hierarchical planner has been used to generate building instructions at different levels of abstraction in Minecraft [Köhn *et al.*, 2020]. In the ultralight domain, the generation of strategic plans using hierarchical planners has been demonstrated in [Jamakatel and Kiam, 2024].

This paper proposes a combination of active and passive context inference for generating contextual strategic plans. Inferring the context using a dialogue with the human in the loop requires that the intents as well as the entities from human utterances can be efficiently extracted. This relates to the task of joint intent detection and slot-filling, a well-studied problem in natural language understanding. We refer to a recent survey [Weld *et al.*, 2022] for an overview of different approaches. The BERT [Devlin *et al.*, 2018] language model has been demonstrated to perform well in joint intent detection and slot-filling tasks [Chen *et al.*, 2019]. Another transformer-based, widely-used language understand-

[10]In practice, the communication happens through speech signals, but we use this configuration to minimize the errors induced by ASR model for comparison.

[11]Asking more questions than specifically required in an emergency scenario is not desirable.

ing model is DIET [Bunk *et al.*, 2020]. Such models are trained on large corpora of domain-specific data, however, in certain domains, such large corpora of data are not available. Recently, the use of synthetic data has gained popularity and synthetic text generation for text classification data has been explored by various authors in literature [Li *et al.*, 2023; Yue *et al.*, 2022; Aggarwal *et al.*, 2023]. An approach of diverse yet accurate text data generation with large language models and human interventions is introduced in [Chung *et al.*, 2023]. An LLM fine-tuned with aviation data has been presented in [Wang *et al.*, 2023]; similarly, an LLM-based Air Traffic Controller agent has been introduced in [Abdulhak *et al.*, 2024].

Besides the task of NLU, for a dialogue system whose primary interface is through audio, a near real-time ASR system is needed. Recently, Whisper [Radford *et al.*, 2022], a general-purpose speech recognition model, has shown state-of-the-art results, and efforts have been made to increase the inference speed of the model [Gandhi *et al.*, 2023] with robust results. The reverse procedure, i.e. the conversion of text to speech, is known as speech synthesis and several data-driven approaches have been developed [Tatanov *et al.*, 2021; Kim *et al.*, 2021]. The dialogue is executed by the underlying dialogue system. End-to-end goal-directed dialogue systems are trained on large-scale datasets [Zhang *et al.*, 2019; Xu *et al.*, 2020]. These systems are, however, bounded by the quality of the training data and often lack transparency about their decisions, making them unsuitable for domains where transparency is crucial. Finite-state dialogue managers, e.g. [Rojas-Barahona and Giorgino, 2009], are usually hand-crafted and require higher effort for creating and updating and are thus not easily maintainable. Various solutions for dialogue agents like IBM's Watson Assistant[12], Google's Dialogflow[13], Amazon's Lex[14] and Microsoft's Bot Framework[15] exist. However, these systems require conversation paths to be explicitly captured in dialogue trees or graphs, making them hard to extend and maintain [Muise *et al.*, 2019]. An approach based on FOND-Planning, where instead of creating and maintaining dialogue graphs, implicit dialogue graphs are compiled from declarative behaviour description, is introduced in [Muise *et al.*, 2019]. Declarative syntax to describe mixed-initiative dialogue has also been presented in [Steedman and Petrick, 2007]. An open-source declarative dialogue planning framework called Plan4Dial has been introduced in [De Venezia and Muise, 2023]. A survey of a plan-based approach for dialogue management can be found in [Santos Teixeira and Dragoni, 2022].

7 Conclusion and Limitations

In this paper, we introduced a neuro-symbolic approach for goal-directed dialogue design by combining planning-based approaches for dialogue execution and strategic plan generation with a neural AI-based method for data augmentation and natural language understanding. Rather than relying solely on LLMs for generating end-to-end goal-directed dialogue, we use dialogue planning to execute verifiable conversations while exploiting the generative capabilities of LLMs to complement scarce human-generated data for the NLU system. We validate the proposed approach in a safety-critical domain of ultralight aircraft for context inference. We showed that generated data for NLU after verification from the human-domain expert can result in similar data diversity metrics. Since the dialogue agent created using this approach is guaranteed to reach the goal state by design, such a dialogue agent is comparable to rule-based dialogue systems. This approach is, however, easier to maintain and expand because of the declarative syntax used for the design. We also demonstrated that in a safety-critical domain, current LLMs are unable to deliver the required results, rendering them unsuitable. Our neuro-symbolic approach exhibits transparency on the flow of the dialogue; its decisions could be explained (if required) and the dialogue be tailored by domain experts, which is essential for human-centred AI.

One key limitation of this approach is that if the local conversation goal changes mid-conversation, the dialogue agent can be stuck in the loop (currently solved by resetting the conversation context variables) because the outcomes expected by the dialogue agent do not correspond to the action realization. In future works, an approach to adjust the dialogue paths to the goal mid-conversation should be implemented without requiring re-planning or restarting the dialogue. Furthermore, the natural language understanding capability of the dialogue can be augmented by using abstract meaning representation. The capability of generating the declarative dialogue syntax using LLMs should also be explored as a way to accelerate the dialogue design and the domain should be expanded to entail all possible emergency scenarios. This work focuses on the system-level design; the usability of the designed system in a realistic and continuous mission scenario still needs to be evaluated.

Ethical Statement

In safety-critical collaborative settings where intelligent systems assist humans, the provided assistance should be verifiable by domain experts. There should be guarantees concerning the system's functionality and soundness. The planning-based approach can be used to achieve these desired properties. The NLU system, which plays a central role in our dialogue system, relies on human and synthetic data. This data has to be carefully examined and verified by the dialogue agent designer so that various edge cases can be accounted for, since unaccounted edge cases could have disastrous consequences. The proposed system will pave the way for reliable, verifiable dialogue agents in safety-critical domains if proper safety measures are taken during data curation.

Acknowledgments

This work is funded by the German Federal Ministry of Economic Affairs and Climate Action (Project MOREALIS).

[12]https://www.ibm.com/cloud/watson-assistant/
[13]https://cloud.google.com/dialogflow/
[14]https://aws.amazon.com/lex/
[15]https://dev.botframework.com/

References

[Abdulhak *et al.*, 2024] Sinan Abdulhak, Wayne Hubbard, Karthik Gopalakrishnan, and Max Z. Li. CHATATC: Large Language Model-Driven Conversational Agents for Supporting Strategic Air Traffic Flow Management. *ArXiv e-prints*, February 2024.

[Aggarwal *et al.*, 2023] Karan Aggarwal, Henry Jin, and Aitzaz Ahmad. ECG-QALM: Entity-controlled synthetic text generation using contextual Q&A for NER. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5649–5660, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Anaby-Tavor *et al.*, 2020] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do Not Have Enough Data? Deep Learning to the Rescue! *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7383–7390, April 2020.

[Behnke *et al.*, 2020] Gregor Behnke, Pascal Bercher, Matthias Kraus, Marvin Schiller, Kristof Mickeleit, Timo Häge, Michael Dorna, Michael Dambier, Dietrich Manstetten, Wolfgang Minker, et al. New developments for robert–assisting novice users even better in diy projects. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 343–347, 2020.

[Brown and Yule, 1983] Gillian Brown and George Yule. *Discourse analysis*. Cambridge university press, 1983.

[Bunk *et al.*, 2020] Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. DIET: lightweight language understanding for dialogue systems. *CoRR*, abs/2004.09936, 2020.

[Chen *et al.*, 2019] Qian Chen, Zhu Zhuo, and Wen Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.

[Chung *et al.*, 2023] John Joon Young Chung, Ece Kamar, and Saleema Amershi. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. *arXiv preprint arXiv:2306.04140*, 2023.

[Clark *et al.*, 2019] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, et al. What makes a good conversation? challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12, 2019.

[Cox *et al.*, 2021] Samuel Rhys Cox, Yunlong Wang, Ashraf Abdul, Christian Von Der Weth, and Brian Y. Lim. Directed diversity: Leveraging language embedding distances for collective creativity in crowd ideation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–35, 2021.

[De Venezia and Muise, 2023] Rebecca De Venezia and Christian Muise. Plan4dial: A dialogue planning framework. 2023.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Gandhi *et al.*, 2023] Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling, 2023.

[Geffner and Bonet, 2013] H. Geffner and B. Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[Guan *et al.*, 2023] Yanchu Guan, Dong Wang, Zhixuan Chu, Shiyu Wang, Feiyue Ni, Ruihua Song, Longfei Li, Jinjie Gu, and Chenyi Zhuang. Intelligent virtual assistants with llm-based process automation. *arXiv preprint arXiv:2312.06677*, 2023.

[Höller *et al.*, 2016] Daniel Höller, Gregor Behnke, Pascal Bercher, and Susanne Biundo. Assessing the expressivity of planning formalisms through the comparison to formal languages. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 26, pages 158–165, 2016.

[Höller *et al.*, 2020] Daniel Höller, Gregor Behnke, Pascal Bercher, Susanne Biundo, Humbert Fiorino, Damien Pellier, and Ron Alford. HDDL: An extension to PDDL for expressing hierarchical planning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9883–9891, 2020.

[Honecker and Schulte, 2017] Fabian Honecker and Axel Schulte. Automated online determination of pilot activity under uncertainty by using evidential reasoning. In *Engineering Psychology and Cognitive Ergonomics: Cognition and Design: 14th International Conference, EPCE 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II 14*, pages 231–250. Springer, 2017.

[Jamakatel and Kiam, 2024] Prakash Jamakatel and Jane Jean Kiam. A system level overview of FRICO – a single-pilot cockpit assitance system. IEEE International Conference on Human-Machine Systems(ICHMS), May 2024.

[Jamakatel *et al.*, 2023] Prakash Jamakatel, Pascal Bercher, Axel Schulte, and Jane Jean Kiam. Towards intelligent companion systems in general aviation using hierarchical plan and goal recognition. In *International Conference on Human-Agent Interaction, HAI 2023, Gothenburg, Sweden, December 4-7, 2023*, pages 229–237. ACM, 2023.

[Kambhampati, 2024] Subbarao Kambhampati. Can LLMs Really Reason and Plan? – Communications of the ACM, March 2024. [Online; accessed 1. Mar. 2024].

[Kiam and Jamakatel, 2023] Jane Jean Kiam and Prakash Jamakatel. Can HTN planning make flying alone safer? In *Proceedings of the 6th ICAPS Workshop on Hierarchical Planning (HPlan 2023)*, pages 44–48, 2023.

[Kim *et al.*, 2021] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. *ArXiv e-prints*, June 2021.

[Köhn *et al.*, 2020] Arne Köhn, Julia Wichlacz, Álvaro Torralba, Daniel Höller, Jörg Hoffmann, and Alexander Koller. Generating instructions at different levels of abstraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2802–2813, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[Kojima *et al.*, 2022] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[Kumar *et al.*, 2019] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and Wlliam Campbell. A closer look at feature space data augmentation for few-shot intent classification. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 1–10, Hong Kong, China, November 2019. Association for Computational Linguistics.

[Larsson and Traum, 2000] S. Larsson and D. R. Traum. Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering*, 6(3):323–340, January 2000.

[Li *et al.*, 2001] Guohua Li, Susan P Baker, Jurek G Grabowski, and George W Rebok. Factors associated with pilot error in aviation crashes. *Aviation, space, and environmental medicine*, 72(1):52–58, 2001.

[Li *et al.*, 2023] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*, 2023.

[Muise *et al.*, 2012] Christian Muise, Sheila McIlraith, and Christopher Beck. Improved Non-Deterministic Planning by Exploiting State Relevance. *Proceedings of the International Conference on Automated Planning and Scheduling*, 22:172–180, May 2012.

[Muise *et al.*, 2019] Christian Muise, Tathagata Chakraborti, Shubham Agarwal, Ondrej Bajgar, Arunima Chaudhary, Luis A. Lastras-Montano, Josef Ondrej, Miroslav Vodolan, and Charlie Wiecha. Planning for Goal-Oriented Dialogue Systems. *ArXiv e-prints*, October 2019.

[Onorati *et al.*, 2023] Teresa Onorati, Álvaro Castro-González, Javier Cruz del Valle, Paloma Díaz, and José Carlos Castillo. Creating personalized verbal human-robot interactions using llm with the robot mini. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 148–159. Springer, 2023.

[Radford *et al.*, 2022] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[Rojas-Barahona and Giorgino, 2009] L. M. Rojas-Barahona and T. Giorgino. Adaptable dialog architecture and runtime engine (AdaRTE): A framework for rapid prototyping of health dialog systems. *International Journal of Medical Informatics*, 78:S56–S68, April 2009.

[Santos Teixeira and Dragoni, 2022] Milene Santos Teixeira and Mauro Dragoni. A review of plan-based approaches for dialogue management. *Cognitive Computation*, 14(3):1019–1038, 2022.

[Steedman and Petrick, 2007] Mark Steedman and Ronald Petrick. Planning dialog actions. In Harry Bunt, Simon Keizer, and Tim Paek, editors, *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 265–272, Antwerp, Belgium, September 2007. Association for Computational Linguistics.

[Tatanov *et al.*, 2021] Oktai Tatanov, Stanislav Beliaev, and Boris Ginsburg. Mixer-TTS: non-autoregressive, fast and compact text-to-speech model conditioned on language model embeddings. *ArXiv e-prints*, October 2021.

[Valmeekam *et al.*, 2023] Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. On the planning abilities of large language models (a critical investigation with a proposed benchmark). *arXiv preprint arXiv:2302.06706*, 2023.

[Wang *et al.*, 2023] Liya Wang, Jason Chou, Xin Zhou, Alex Tien, and Diane M Baumgartner. Aviationgpt: A large language model for the aviation domain. *arXiv preprint arXiv:2311.17686*, 2023.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv e-prints*, January 2022.

[Weld *et al.*, 2022] Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys*, 55(8):1–38, 2022.

[Xu *et al.*, 2020] Haotian Xu, Haiyun Peng, Haoran Xie, Erik Cambria, Liuyang Zhou, and Weiguo Zheng. End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization. *World Wide Web*, 23:1989–2002, 2020.

[Yue *et al.*, 2022] Xiang Yue, Huseyin A Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Synthetic text generation with differential privacy: A simple and practical recipe. *arXiv preprint arXiv:2210.14348*, 2022.

[Zhang *et al.*, 2019] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019.