Plan4Dial: A Dialogue Planning Framework

Rebecca De Venezia, Christian Muise

Queen's University, Canada

{ 18rldv, christian.muise} @ queensu.ca

Abstract

Dialogue agents have exploded in importance in recent years as businesses increasingly use chat-bots to serve their customer base. However, many of these dialogue systems rely on black-box language models that cannot be verified for predictability, making them a liability. One proposed solution is dialogue planning, which uses planning to generate a complete dialogue tree and allows for the verification of the agent's actions. Despite the plethora of existing research in this space, there is no open and readily available modern framework for dialogue planning development. We propose Plan4Dial, an open-source system for creating dialogue planning chat-bots. Plan4Dial allows developers to declare complex chat-bots with ease by writing an intuitive YAML specification which the system converts to raw PDDL. Plan4Dial then calls a state-of-the-art planner to generate a dialogue tree which we execute with an extension of IBM's dialogue plan executor, Hovor. We also created WIDGET, an embeddable web user interface for users to chat with their agents. Our work allows for the simple creation of complex but verifiable goal-oriented dialogue agents using planning technology.

System + Related Work: github.com/dialogue-planning

Demo: youtu.be/8BoqfhW5PEQ

1 Introduction

Motivation The market for chat-bots was valued at 5.1 billion USD in 2022 and is predicted to grow over the next decade (Grand View Research Inc. 2023), especially with the advancement of large language models (Shahriar and Hayawi 2023). Although these models have made huge strides in NLP-related tasks, they are unfit for goal-oriented conversations by themselves as they often struggle with reasoning (Valmeekam et al. 2023). Other approaches have improved predictability with hand-drawn dialogue trees, but these are difficult to scale (Muise et al. 2019). With chat-bots being deployed to a wider variety of domains, the risk for liability increases unless the agents can be verified to reach their goals. Dialogue planning aims to address these concerns.

Dialogue Planning Dialogue planning is a framework for creating chat-bots that uses automated planning to synthesize a dialogue tree. The approach ensures the agent's actions are fully explainable while forgoing the unpredictable results of end-to-end neural solutions and the need to enumerate complex trees (Muise et al. 2019).

Listing 1: Dialogue as Planning Example PDDL

```
(:action get-size
    :parameters()
    :precondition
        (and
            (not (know__size))
            (not (force-fallback))
        )
    :effect
        (labeled-oneof set-size
            (outcome valid
                (and (know__size))
            )
            (outcome fallback
                (and (force-fallback))
            )
        )
)
```

In Listing 1, we observe an action taken from a pizza-ordering chat-bot that asks the user for their preferred size. The preconditions are that 1) the agent does not know the size yet and that 2) a fallback action (which handles erroneous user input) is not occurring. Next, FOND (fully observable non-deterministic planning) allows the specification of multiple outcomes where one is selected at runtime. Here, we have one outcome where the agent learns the size and another that forces a fallback to handle invalid input. The PDDL model is an abstraction of the conversation as the planner generally only needs to know what the agent *knows* without the details (i.e. "medium") to drive agent behaviour (Muise et al. 2019). Finally, a FOND planner based off PRP (Muise, McIlraith, and Beck 2012) generates a plan.

Dialogue planning does not abandon neural AI entirely, as a language model is still used to extract user input. Thus, the framework harnesses the advantages of neural and symbolic AI to create a powerful approach for executing accurate yet verifiable conversations. In 2019, IBM released D3WA, a system for designing goal-oriented agents, and Hovor, a pro-

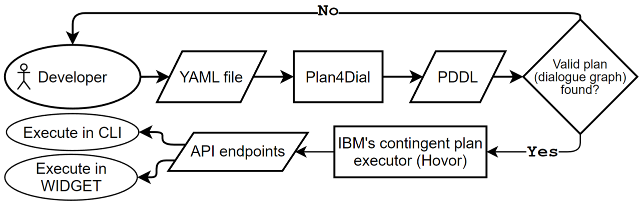

Figure 1: Flow of the Framework.

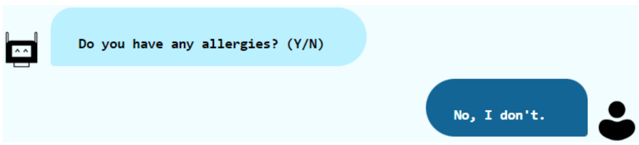

Figure 2: WIDGET (Web Interface for Dialogue agents that are Goal-oriented and Enabled Through planning).

gram that executes these plans. Our work replaces D3WA for Plan4Dial and builds off Hovor to ease the creation of agents and increase their complexity.

2 A Framework for Dialogue Planning

Figure 1 details the design flow of our approach. The developer writes a YAML file which contains a declarative description of their agent (Figure 3). The file is then passed to Plan4Dial, which converts the file to raw PDDL and attempts to generate a plan. Note that Plan4Dial prevents the need to write raw PDDL, which becomes complex for many action types. The valid plan, if found, is then passed to our extension of Hovor to execute the conversation, either through the CLI or in our embeddable web UI (Figure 2).

2.1 Plan4Dial

The YAML file passed to Plan4Dial allows for a range of complex capabilities but uses a simple structure comprised of three main subsections:

- **Context variables** allow the agent to store entities (information extracted from user input) and details about the current state.
- **Intents** indicate what the user is attempting to accomplish with their utterance. Every action outcome has a corresponding intent.
- **Actions** are the driving force of the conversation and define what the agent can do and when.

An example of each is defined in Figure 3. The action corresponds to the PDDL in Figure 1. Next, Plan4Dial makes the following improvements on IBM's D3WA:

- Includes tools to create custom action templates which streamline agent declaration. One example included with the software is `slot_fill`, which specifies an action that extracts any number of entities from the user and can even "clarify" entities extracted with low confidence.
- Allows the developer to specify how entities are extracted. Rasa's off-the-shelf NLU model (Bocklisch et al. 2017) is used as the default and is trained on the `context_variables` and `intents` in the YAML file. The user can also specify Rasa NLU plugins, like Spacy (Honnibal and Montani 2017), or a regex checker.
- Is open source and easily extensible.

```yaml
context_variables:
  size:
    type: enum
    known:
      type: flag
      init: false
    options:
      - small
      - medium
      - large
intents:
  share_size:
    variables:
      - size
    utterances:
      - I want a $size pizza.
      - I would prefer a $size pizza.
      - Can I get a $size pizza?
actions:
  get-size:
    type: dialogue
    message_variants:
      - What size do you want your pizza to be?
    condition:
      size:
        known: false
    effect:
      set-size:
        oneof:
          outcomes:
            valid:
              updates:
                size:
                  value: $size
                  known: true
              intent: share_size
```

Figure 3: Plan4Dial Example YAML. A "$" denotes the extracted value of the variable, i.e. "medium."

2.2 Extensions on Hovor

IBM's Hovor is responsible for executing the conversation and determining the outcome of an action based on user input. We extended Hovor by creating more complex outcome determiners, which allows for richer actions. For example, our "Context dependent" outcome determiner only accepts an outcome if a condition for that outcome holds (e.g., if "size" is "small"). The developer can also create custom outcome determiners.

3 Summary

In summary, we propose a framework that streamlines the creation of complex goal-oriented dialogue agents and makes dialogue planning accessible to the community.

Acknowledgements

We gratefully acknowledge funding from the Undergraduate Student Research Awards from the Natural Sciences and Engineering Research Council of Canada.

References

Bocklisch, T.; Faulkner, J.; Pawlowski, N.; and Nichol, A. 2017. Rasa: Open Source Language Understanding and Dialogue Management. arXiv:1712.05181.

Grand View Research Inc. 2023. Chatbot Market Size, Share & Trends, Analysis Report By Application (Customer Services, Branding & Advertising), By Type, By Vertical, By Region (North America, Europe, Asia Pacific, South America), And Segment Forecasts, 2023 - 2030.

Honnibal, M.; and Montani, I. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Muise, C.; Chakraborti, T.; Agarwal, S.; Bajgar, O.; Chaudhary, A.; Lastras-Montano, L. A.; Ondrej, J.; Vodolan, M.; and Wiecha, C. 2019. Planning for Goal-Oriented Dialogue Systems. arXiv:1910.08137.

Muise, C.; McIlraith, S.; and Beck, C. 2012. Improved Non-Deterministic Planning by Exploiting State Relevance. *Proceedings of the International Conference on Automated Planning and Scheduling*, 22(1): 172–180.

Shahriar, S.; and Hayawi, K. 2023. Let's have a chat! A Conversation with ChatGPT: Technology, Applications, and Limitations. arXiv:2302.13817.

Valmeekam, K.; Sreedharan, S.; Marquez, M.; Olmo, A.; and Kambhampati, S. 2023. On the Planning Abilities of Large Language Models (A Critical Investigation with a Proposed Benchmark). arXiv:2302.06706.