```
class DialogueAgent(Agent):
   <fields>
   def handle_turn(self, prev_belief_state: BeliefState,
                    last system utterance: str, last system acts: List[Act],
                    user_utterance: str, user_intent: List[Intent],
                    db result: List[Dict[str, str]] = None,
                    system acts: List[Act] = None,
                    system response: str = None):
       <fake implementation>
   <one method per intent, with all of its informable arguments and</pre>
    their descriptions derived from the schema. book_taxi is kept below
    as an example. An additional method indicating no intent is included
    called `no_change()`>
   @abc.abstractmethod
   def book taxi(self, leave at: str = None, destination: str = None,
                  departure: str = None,
                  arrive by: str = None) -> Intent:
        book taxis to travel between places
        Parameters:
        leave_at: str
            leaving time of taxi
       destination: str
            destination of taxi
       departure: str
            departure location of taxi
        arrive by: str
            arrival time of taxi
        pass
if __name__ == '__main__':
   agent = DialogueAgent()
   # Provide a user utterance matching the formal user intent
   # Format integers as `int`s, not strings. String values do not need
   # underscores for spaces. Put time values in 'hh:mm' format
   <in-context exemplars from self-predictions may go here>
   # Example 1
   response = agent.handle turn(
        belief state=BeliefState(hotel=dict(price range="cheap",
                                            internet="yes",
                                            area="south")),
       last system utterance="Yes, that would be Rosa's B & B. " + \
                              "Would you like me to book you a room?",
        user_intent=[agent.book_hotel(people=5, name='rosas b & b')],
        user_utterance="Yes, please book a room for 5 people.",
```