Managing Emotional Dialogue for a Virtual Cancer Patient: A Schema-Guided Approach

Benjamin Kane, Catherine Giugno, Lenhart Schubert, Kurtis Haut, Caleb Wohn, and Ehsan Hoque

Abstract—In this paper, we describe a general-purpose dialogue management framework used to design SOPHIE (Standardized Online Patient for Healthcare Interaction Education). SOPHIE simulates a virtual standardized cancer patient that allows physicians to practice skills such as empathy and patient empowerment in end-of-life communication. To provide the user with an opportunity to practice these skills, SOPHIE must produce a natural, emotionally appropriate conversation, yet handle topic shifts and open-ended questions from the user. To accomplish this, our approach to dialogue management loosely follows schemas – explicit representations of the typical flows of dialogue in end-of-life communication – while also using flexible pattern-driven methods for interpretation and generation. We conduct a crowdsourced evaluation of conversations between medical students and SOPHIE. Our agent is judged to produce responses that are natural, emotionally appropriate, and consistent with her role as a cancer patient. Furthermore, it significantly outperforms an end-to-end neural model fine-tuned on a human standardized patient corpus, attesting to the advantages of a schema-guided approach in this domain. However, the system is currently limited in its ability to generate responses that are judged to demonstrate deep understanding of the user, suggesting that future work should place focus on integrating this framework with robust natural language understanding and commonsense reasoning methods.

Index Terms—Virtual agent, Standardized virtual patient, Patient-physician communication, Dialogue, Schema-guided

1 INTRODUCTION

CONVERSATIONAL virtual standardized patients (VSPs) – virtual humans that simulate patient interactions for use in training or evaluating medical practitioners – present an impactful but challenging application of affective dialogue technology. Communication skills on the part of the physician are a well-recognized determinant of patient satisfaction and optimal health outcomes [1], [2], [3], [4], [5], [6]; yet providing communication training for physicians is a resource-intensive process – requiring extensive training for the role and taxing emotional investment from patient actors. As a result, the quality and availability of communication training are often quite variable. The ability to create, customize, and deploy VSPs that are capable of emulating realistic patient conversations with physicians could have a broad impact on medical training.

In previous work, we introduced a novel VSP system, SOPHIE (Standardized Online Patient for Health Interaction Education), aimed at improving patient-physician communication in cancer care by providing automated feedback on the physician’s communication skills – such as their ability to be empathetic towards SOPHIE’s emotions in discussing end-of-life prognosis and care [7], [8]. While the overall system was reviewed favorably in a qualitative user study, the dialogue manager used was rather preliminary and limited in its capabilities, which was reflected in user feedback – such as that SOPHIE often shifted topics prematurely.

The utility of automated feedback metrics for behavior such as empathy, however, depend critically on the quality of dialogue supported by the system – a VSP must provide the physician with opportunities to be empathetic (or to empower the patient, etc.), by generating natural reactions that are consistent with a real patient in their setting, express appropriate emotions, and demonstrate understanding of the physician’s responses. Effective dialogue management in this domain is challenging because the types of conversations that oncologists have with real cancer patients are emotionally and topically diverse, and often involve complex mixed-initiative dialogue – i.e., a mixture of the patient taking initiative and the physician taking initiative at various times. These types of dialogues have radically different structure than task-oriented dialogues, requiring a greater degree of collaboration between the two agents [9]. Furthermore, these dialogues can span a relatively wide range of topics, and can jump suddenly between topics.

Owing to present limitations in dialogue management technology, current VSPs have been largely limited to narrow single-initiative domains – for instance, allowing a doctor to practice eliciting a medical history from a patient and creating a differential diagnosis, where the interaction is enabled using a combination of statistical retrieval and pattern-matching methods [10], [11], [12], [13]. Other more general techniques, such as fine-tuning large language models on domain corpora, have been successfully employed for creating robust conversational chatbots for entertainment purposes [14], [15], [16]. Yet, these methods present limitations for developing realistic VSPs such as a lack of goal-directed planning abilities and a tendency to “hallucinate” false or contradictory information [15]. To our knowledge, there is no dialogue manager that has yet been developed capable of supporting realistic VSPs in open-ended and mixed-initiative patient-physician conversations, such as we aim to support in the SOPHIE system.
In this paper, we present and evaluate a novel schema-guided approach to dialogue management used to improve the conversational abilities of SOPHIE, allowing for more robust control over topical shifts and emotionally appropriate response generation. Schemas encode explicit knowledge of the typical flow of dialogue in a particular domain, while allowing “on-the-fly” modification as the conversation proceeds. A schema-guided approach has been shown effective for the design of task-specific dialogue systems [17], [18], [19] as well as friendly peer systems capable of casual conversation across many topics [20]. We extend this approach to support natural dialogue management for the SOPHIE VSP, designing dialogue schemas for various topics that typically occur in patient-physician end-of-life communication, in collaboration with palliative care experts. Furthermore, the proposed dialogue management architecture is general and can be used as a platform to create other virtual humans, requiring only the creation of a set of schemas and pattern-matching rules for extracting and responding to explicit, portable “gist clauses” from user input.

Our specific contributions in this paper include:

1) A general-purpose dialogue manager based on dialogue schemas and a unique method of “gist clause” extraction as a preliminary step in mapping user utterances to semantic representations.

2) The development of natural, thematically consistent, and emotionally appropriate dialogue capabilities for the SOPHIE virtual cancer patient, developed in collaboration with palliative care experts.

3) A crowdsourced evaluation of 30 dialogue transcripts obtained in a pilot study of the SOPHIE VSP, demonstrating that the VSP’s conversational performance was found to be fluent and natural, emotionally appropriate, and consistent with her role as a cancer patient, albeit currently limited in its ability to demonstrate true understanding of the preceding conversation.

4) A comparison of the SOPHIE VSP to a large language model fine-tuned on a corpus of human standardized patient (SP) data from our domain – demonstrating that an intentionally designed, schema-guided approach to dialogue management is able to outperform an end-to-end neural approach in producing natural and emotionally appropriate reactions for end-of-life communication (though we discuss ongoing work in hybridizing these approaches to enable more robust generation).

While these systems were engineered for particular domains, other approaches to dialogue system design aimed to create generic dialogue shells for collaborative problem-solving that could be extended to particular domains [26], [27], [28]. These systems were primarily oriented towards task-specific domains, but do not generalize well to meaningful open-ended conversation where emulating more subtle behavior – such as showing empathy – typically requires knowledge of social conventions rather than systematic planning. More recently, end-to-end neural language models have been effective in creating chatbots for entertainment purposes [14], [15], though the “black box” nature of these models has so far precluded their use in sensitive domains.

To bridge the gap between classical dialogue management techniques (those based on symbolic knowledge and collaborative planning) and end-to-end neural dialogue models, the paradigm of schema-guided dialogue has been proposed. This approach allows for flexible management of dialogue, through matching of observed utterances to explicitly specified dialogue policies or “schemas”. Contemporary work on schema-guided dialogue has shown that robust zero-shot dialogue behavior could be achieved by fine-tuning language models to fill in schemas consisting of slot-value pairs [17] or to align schemas consisting of conversation graphs with user utterances [18], [19]. In contrast, we employ a schema-guided approach in an open-ended VSP domain as opposed to task-specific dialogues, and rely on a semantically rich event schema representation instead of slot-value pairs or conversation graphs.

The ensembling of symbolic and neural AI techniques has been shown to be an effective strategy for many applications of affective computing – particularly for sentiment analysis, where some recent work has utilized specialized commonsense knowledge bases and reasoning capabilities for detecting affect in text [29], [30], [31]. In this paper, we focus on the complementary task of generating affective responses using dialogue schemas, a semantic representation compatible with both symbolic and neural techniques.

## 2 Related Work

### 2.0.1 Approaches to Affective Dialogue Management

The design of virtual humans for therapeutic use and conversational coaching has become a very active research area. Some of the earliest systems, such as the “How Was Your Day” virtual human [21] and the ERICA android [22], used a template-based approach to generate empathetic responses in casual conversations. The SimSensei virtual human [23] uses a classifier-based approach to provide healthcare support to users and detect potential PTSD, while the INOTS virtual human [24] uses a similar approach to coach naval officers on interpersonal skills. The LISSA virtual human [20], [25] allows users to practice social skills through casual conversation using a schema-guided framework.

Other virtual patient research has focused on enabling more natural dialogue behavior for VSPs across several different domains, typically through the use of pattern-matching and classifier-based methods [10], [11], [12], [13], [33], [34]. However, these systems are generally designed around single-initiative domains, such as history-taking and diagnostic conversations (in which the physician has initiative and needs to solicit information from the patient); to our knowledge, there is no VSP system that can support open-ended mixed-initiative dialogue such as that of a cancer patient discussing end-of-life care with a physician.
3 DIALOGUE MANAGER OVERVIEW

In this section, we provide an overview of the Eta dialogue manager used to develop the SOPHIE VSP. Eta is a general-purpose, modular, schema-based dialogue management framework, diagrammed in Figure 1. Eta is built upon previous work on the LISSA system [35], and like LISSA, the Eta dialogue manager relies centrally on a database of dialogue schemas – representations of prototypical dialogues – to flexibly plan dialogues, as well as transduction trees for interpretation and response generation. In contrast to the LISSA system, Eta is a general-purpose dialogue shell that supports a wider range of behavior, and can be used to develop virtual humans in multiple different domains. Whereas LISSA was largely guided by a predetermined order of utterances and user responses, Eta is capable of alternating between multiple parallel tasks including observation, interpretation, schema instantiation/execution, and reasoning. In order to create a virtual human using Eta, a designer need only create a set of dialogue schemas and transduction trees; many aspects of these trees are portable across domains. We discuss the design of the SOPHIE virtual human in Section 4.

3.1 Dialogue Schemas

The concept of a “schema” has a rich history in AI research, arising from attempts to capture generic and prototypical knowledge in a declarative symbolic form [36], [37], [38]. A schema is typically conceptualized as a packet of expected properties of an object or event, such that a strong match between an observed object and properties of a schema allows one to make inferences about other properties associated with that schema. Event schemas also represent a generalization of hierarchical plans in the sense that they can be used for classic planning through goal chaining and step decomposition, but also more “reflexive” planning based on expectation-matching and abductive inference [39].

The Eta dialogue manager relies on a database of dialogue schemas: event schemas for various dialogues that an agent might encounter. Formally, our schema representation is based on Episodic Logic (EL) – a type-coherent logical semantic representation that closely matches the form and expressive capacity of natural English. A dialogue schema minimally consists of a series of expected episodes (assumed to occur sequentially by default, though other temporal orderings are possible), but may also contain schema sections containing EL formulas for other expected properties of the dialogue – participant roles, preconditions, goals, etc. Episodes within a dialogue schema may correspond to expected user inputs, observations, primitive actions that the agent is capable of executing, or complex actions/events that themselves correspond to some event schema.

The behavior of the dialogue manager is driven by the selection and incremental instantiation of a dialogue schema from its schema database (indicated by arrow 1 in Figure 1), beginning with a top-level schema for the overall dialogue. An example of a dialogue schema from the SOPHIE VSP (representing a “subdialogue” in which the patient asks about their prognosis) is shown in Figure 2. Symbols beginning with ‘?’ represent variables that are replaced with constants upon instantiation of a schema, while ‘‘me’’ and ‘‘you’’ indexically refer to the dialogue participants. The schema begins with a header that provides a characterizing formula for the head episode ‘?e’, and is followed by a schema section specifying goals for that episode. The body of the schema specifies that the subdialogue consists of a sequence of episodes: The agent asks the user some variant of the quoted question; the user replies to the agent, and the

1. See [40] for a full list of supported schema sections.
3.2 Dialogue Context and Episodic Memory

The various components of Eta rely on a central database divided into a dialogue context (containing facts in the common ground, assumed to be “true now”), and an episodic memory containing knowledge of both events that occurred during the course of the conversation, as well as events in the system’s “self-knowledge” (e.g., that SOPHIE had undergone radiation therapy previously). These facts are represented and stored as EL formulas, indexed on their predicates and arguments for efficient retrieval.

3.3 Schema Instantiation

The dialogue manager maintains a pointer to the currently pending episode in its partially instantiated schema (as well as a pointer to the previous episode, next episode, and a subschema, when applicable), and attempts to instantiate this episode at each time step.

If the currently pending episode is an expected user action, or some external event, the system attempts to match that expectation to a “currently true” fact in context (indicated by arrow 2 in Figure 1), unifying any variables throughout the schema. If the currently pending episode is a system action, the system will execute that action – either selecting and expanding the appropriate subschema, if the action is non-primitive, or otherwise invoking an executable function of the system, which may involve updating dialogue context, domain-specific reasoning, and/or response generation (indicated by arrows 3, 7, and 9 in Figure 1).

When an episode in a schema is successfully instantiated, the corresponding episode constant in episodic memory is substituted for the episode variable in the schema. For example, Figure 1 shows a state in which “?e1” from the schema in Figure 2, involving SOPHIE paraphrasing a question about her prognosis to the doctor, has been successfully instantiated and expanded into a primitive “say-to” action. This instantiation resulted in the addition of the episode constant E1 to episodic memory, characterized by the “paraphrase-to” and “say-to” formulas. Consequently, E1 is substituted for the episodic variable ?e1 throughout the partially instantiated schema.

Since episodes in schemas only represent expectations about typical dialogues, it is entirely possible that the dialogue manager will not be able to match any observation to an expected episode. Each episode in a schema is associated with a probabilistic certainty indicating the strength of this expectation. In the case of a failure to match an observation, the dialogue manager will use this certainty to determine how many time steps it should wait before skipping that episode and proceeding with the remainder of the schema.

3.4 Perception and Interpretation

Eta supports integration with various perceptual servers handling different modalities, including text interfaces, speech recognition programs, and vision systems. These servers collect observations from the environment and send inputs to the core Eta dialogue manager through EL queries, where they are stored in the system’s episodic knowledge.

In the case where an input corresponds to a natural language utterance (indicated by arrow 4 in Figure 1), the system also attempts to interpret the semantic meaning of the utterance using the context of the immediately preceding speech act in the partially instantiated schema (indicated by arrow 5 in Figure 1). Semantic interpretation is performed at two levels, allowing for both shallow and deep understanding. First, an explicit context-independent gist clause is extracted from the user’s input representing the gist of the user’s utterance. The intuition behind this stage of interpretation is that the context of the immediately preceding turn is typically sufficient to resolve many context-sensitive phenomena (e.g., ellipsis, simple anaphora, ambiguity, etc.) using relatively superficial text processing; mapping a complex utterance to a context-independent ‘canonical’ form greatly simplifies semantic parsing and response generation. In the case of the SOPHIE VSP, this shallow stage of interpretation is often sufficient for response generation.

Optionally, Eta can then parse a gist clause into EL – more specifically, into an Unscoped Logical Form (ULF) representation. This surface-oriented, preliminary form of EL allows for more direct parsing by leaving tense operators, quantifiers and coordinators uns Scoped. This ULF is then scoped and ‘de-indexed” (introducing temporally related event tokens and explicit event relationships) using generic methods to obtain a final representation to add to episodic knowledge (indicated by arrow 6 in Figure 1). A constraint-based reference resolution module also attempts to resolve any references not handled during gist clause interpretation.

3.4.1 Pattern Transduction

In practice, gist clause interpretation is performed using the hierarchical pattern transduction method previously described.

The default certainty of an episode is 1, indicating that the system should wait indefinitely.
introduced by the LISSA system [35], which allows for inexact feature-based matching. Transduction trees specify patterns at their nodes, with branches from a node providing alternative continuations as a hierarchical match proceeds. Terminal nodes have associated directives indicating whether they provide a gist clause template, send input to some subordinate tree, or some other outcome. The pattern nodes use simple template-like patterns that look for particular words or lexical features, and allow for “match-anything”, length-bounded word spans.

An example of hierarchical pattern transduction for gist clause interpretation in the SOPHIE domain is shown in Figure 3 (left). A top-level tree first matches keywords in SOPHIE’s previous gist clause to select a set of relevant subtrees to use to interpret the user’s utterance (shown in the middle) to produce a final gist clause (shown at the bottom). In the case of generation, SOPHIE’s current gist clause (shown at the top) is used to select relevant subtrees to use to produce an utterance (shown at the bottom) using the context of the user’s previous gist clause (shown in the middle). In both cases, the paths chosen in the pattern transduction process are highlighted, as well as relevant parts of the inputs used in matching.

3.5 Reasoning, Inference, and Schema Selection

Eta allows for several methods for reasoning, inference, and the selection of new schemas based on context. Similarly to the perceptual servers, Eta supports integration with reasoning servers capable of specialist reasoning in different domains – including temporal models, spatial models, and type ontologies. These reasoning servers communicate with the core dialogue manager through EL queries, and may be invoked in the process of schema instantiation to infer additional facts from context/memory (indicated by arrows 7 and 8 in Figure 1). Eta also has transduction-based mechanisms for drawing additional pragmatic inferences from gist clauses in the process of interpretation (e.g., in the SOPHIE domain, a response from the doctor that mentions a relative remains future work.

In order to allow for the dynamic modification of the dialogue schema in response to unexpected dialogue trajectories, Eta also supports a few methods for selecting a new schema from its schema knowledge (indicated by arrow 11 in Figure 1), updating or replacing the current partially instantiated schema. First, Eta may react to a reply from the user with a new schema – for instance, in the example in Figure 1 from the SOPHIE domain, if the user...
replies with “What are your goals for the future?” for ?e2, SOPHIE might react with a new schema for telling the user about her goals for the future (for example, to attend her grandson’s graduation). Second, if a subschema contains a goal condition that is already satisfied according to the system’s memory (e.g., if the doctor had already mentioned something about SOPHIE’s prognosis earlier), the system may skip over that subschema and proceed to the next episode in the superordinate dialogue schema. Third, if the episodes of a schema are fully instantiated but the goal still isn’t satisfied, the system can attempt to select a new schema as a fallback strategy (e.g., repeating the question about prognosis, or asking a more specific question). These schema selection strategies are currently determined using pattern transduction trees based on the content of the user reply and/or the relevant schema goals.

3.6 Response Generation
The overall process for response generation mirrors that of semantic interpretation: generation may begin from the level of EL, canonical gist clauses, or surface-level natural language, depending on the complexity of the system action initiating generation (indicated by arrow 9 in Figure 1).

In the case of a simple reaction from the system, hierarchical pattern transduction trees may be used to select a direct natural language output from the system based on the user’s previous gist clause; the terminal nodes of these transduction trees specify templates that may borrow from the content of the matched gist clause.

In more complex system actions, such as “paraphrase-to” actions (in which the system says some variant of the quoted gist clause), the response generation module attempts to form a natural language utterance from a gist clause. Currently, this is also done through hierarchical pattern transduction methods in a manner reflecting interpretation: the user’s previous gist clause will be used as context for paraphrasing the system’s gist clause into a more natural form. An example pattern transduction for response generation in the SOPHIE domain is shown in Figure 3 (right). The directives at leaf nodes specify response templates that may get partially “filled in” by content borrowed from the user’s gist clause – in this case, since the doctor specified an explicit prognosis of three months, SOPHIE expresses denial by questioning the prognosis and relaying an anecdote about a relative who outlived a similar prognosis. Multiple templates may be provided for a single leaf node in order to prevent redundancy or repetition.

Note that these templates may also include emotion tags – e.g., [SAD] – that are not included in the verbalized output, but are used by the virtual human to produce non-linguistic affective behavior, such as prosody and facial expression.

Finally, inferences generated by specialist reasoning servers may be used for response generation (indicated by arrow 10 in Figure 1). In these cases, the response generator receives an EL formula as input, which it then maps to a corresponding gist clause using a generic method. The gist clause is mapped to a surface utterance in the same manner described previously.

4 SOPHIE EVALUATION
4.1 Design of SOPHIE Virtual Human
The SOPHIE virtual human, shown in Figure 1, was designed to represent an elderly lung cancer patient who had had previous testing done, and is meeting with a doctor to learn more about her test results and prognosis. Users interact with the system verbally, with a manually calibrated silence threshold determining turn-taking, and an off-the-shelf automatic speech recognizer (ASR) converting audio to text input. To produce non-linguistic emotional behavior on the part of the virtual human, we also modulate her speech pitch and facial expression based on the emotion tags received from the dialogue manager (see Section 3.6), although in this paper we focus on linguistic behavior.

The dialogue capabilities of the virtual human were enabled through the creation of dialogue schemas for various topics that the virtual human may discuss (such as the example in Figure 2), a top-level dialogue schema containing the expected sequence of topics that the virtual human will ask about in a session (admitting modification during the conversation), and a set of pattern transduction trees for interpretation and response generation. These components were hand-engineered in consultation with palliative care experts; in order to anticipate possible inputs and create realistic responses, we also used a corpus of human SP conversation transcripts from the VOICE dataset, as well as feedback and transcripts from a small focus group study involving medical students interacting with SOPHIE.

Our pattern transduction-based approach allows for accessible customization by dialogue designers who are not AI experts. A non-expert dialogue designer was trained in the pattern transduction language and was able to independently create new transduction trees within a week. On average, they were able to create a transduction tree for a new topic in approximately 15-30 minutes (depending on the level of detail required and whether or not the new topic is related to any previous topics); about 30 such topics were sufficient to achieve the results presented in this paper.

4.2 Evaluation Method
In the following sections, we describe our evaluation of the SOPHIE VSP using both expert and crowdsourced annotations of turns from SOPHIE conversations. We also collect annotations of responses generated by a large language model, fine-tuned on a corpus from our domain, across the same user inputs in order to compare the capabilities of our schema-based dialogue shell to those of an end-to-end neural approach.

4.3 Pilot Dataset
We use the transcripts collected from an initial pilot experiment that evaluated the effects of the SOPHIE system on user communication skills. This experiment consisted...
of 30 medical student participants, half of whom were assigned to a treatment group that interacted with SOPHIE for two sessions each. From these transcripts, we obtained a dataset consisting of 397 conversation turns across 30 conversation sessions from 15 participants.

4.3.1 Expert Annotations
As a preliminary form of evaluation, the researchers involved in the design of the pattern transduction trees for SOPHIE independently provided expert turn-by-turn annotations on whether the system extracted a correct gist clause, an incorrect gist clause, or failed to extract a gist clause; as well as whether the system gave an appropriate response, inappropriate response, or a non-contentful clarification request. They also annotated any notable ASR errors that were observed, which fell into two categories: transcription errors and turn-taking errors (i.e., where the user was cut off).

4.3.2 Neural Baseline
We also establish a “neural baseline” for our conversational domain — that is, the performance of a large language model fine-tuned on human SP transcripts from our domain — and compare the responses generated by this model, when prompted with turns from the pilot dataset, to the responses from our system. Specifically, we used the DialoGPT-medium model [16], as it was the largest language model we were able to train within our financial constraints, and is a standard benchmark for a variety of dialogue tasks. We fine-tuned DialoGPT on the VOICE human SP dataset [41], which contains 109,134 dialogue turns (44,917 of which are patient turns) across 389 dialogues between human actor cancer patients and doctors.

After filtering for only patient turns and creating a 20%/80% training set/validation set split, we trained the model for 5 epochs, using a context window size of 5 previous utterances, and a batch size of 1. This resulted in a validation set perplexity of 6.53; gains after 5 epochs were negligible. To generate model responses for the pilot data, for each turn in the dataset, we concatenated the user utterance with the context of the immediately preceding utterance (separated by end-of-turn tokens) and let the model generate the next response. We used a length penalty of 0.5 and a repetition penalty of 1.4 as generation parameters; these were found to generate the best responses through manual inspection.

4.3.3 Crowdsourced Evaluation
Given the SOPHIE responses and corresponding responses generated by the neural baseline model, we crowdsourced annotations on response quality using Amazon Mechanical Turk. We first removed 89 turns that were judged by either of the expert annotators to include significant ASR errors in the doctor’s input. The remaining turns were used to form 308 items, each consisting of the context of the previous patient utterance, the doctor’s utterance, the response generated by Eta, and the response generated by the neural baseline model.

Items were randomly distributed into 20 Human Intelligence Tasks (HITs), each containing 16 items. To avoid introducing possible annotation bias, items were exactly balanced on string length of each text field (i.e., items were assigned to four high/low bins for each text field computed using a median split), and were approximately balanced on the expert annotations of response quality (ensuring that each HIT had about as many good responses as bad responses from each system). For each item, workers were shown a response from Patient A and the previous two dialogue turns, and then asked to rate the following four questions about the response on a Likert scale. Then workers were shown a response from Patient B to the same previous turns, and asked to rate the same four questions. The Eta responses and neural baseline responses were randomly assigned to Patient A and Patient B.

Q1. Patient A/B’s response is fluent and natural.
Q2. Patient A/B’s response is consistent with her having understood the preceding turns.
Q3. Patient A/B’s response is consistent with her role as a cancer patient.
Q4. Patient A/B’s response expresses appropriate emotions (if an emotional response).

Workers were instructed to focus on the quality of the response irrespective of the patient background and previous turns for answering Q1, and to overlook fluency in the response for answering Q2-Q4. Additionally, Q3-4 had “Not Applicable” options, in the event that these questions could not be evaluated (for instance, Q3 is impossible to evaluate if the patient asks “Could you repeat that?”; and Q4 is impossible to evaluate if the patient says “I’m taking Lortab for the pain.”); any Not Applicable responses were discarded.

4.4 Results
Table 1 shows the averaged expert annotations (N=2) for ASR errors, the fraction of correctly extracted gist clauses, and the fraction of appropriate responses from the system (expected to be higher than the former quantity because the system might fail to extract a gist clause but still give an appropriate “default” response). These statistics are corrected for ASR errors by assuming that a clarification request by the system is the correct behavior when the input contains a significant ASR error and no gist clause is extracted. We also show the fraction of responses judged as appropriate, conditioned on a gist clause having been extracted. Inter-annotator agreement (Cohen’s kappa) was quite high for both annotations, at 0.85 for gist clause annotations and 0.71 for response annotations.

From the crowdsourced Amazon Turk study, we obtained 10 ratings for questions Q1-4 for each item, from 167 native English speakers. The resulting distributions of ratings per item (averaged across participants, discarding ‘N/A’ values) are shown as histograms in Figure 4 (a score of 1 corresponds to “Strongly Disagree”, and 5 to “Strongly Agree”). The mean and median scores for each question across all items are shown in Table 2; the mean differences between Eta and DialoGPT were found to be statistically significant for each question (P < 0.05) using a Mann–Whitney U test.
**4.5 Discussion**

From the expert annotations in Table 1, we observe that the rates of correct gist clause extraction and appropriate response generation were 39% and 49% respectively. In many cases, failure to extract a correct gist clause would also lead to a non-contentful clarification request rather than an appropriate response. However, we observe that in the cases where the system was able to extract a gist clause, the fraction of appropriate responses by the system was significantly higher – suggesting that the primary bottleneck in the current system is the gist clause extraction step.

While the expert-annotated response accuracy is fairly low compared to results from previous VSP evaluations, which report between 60% and 80% response accuracy [10], we argue that our results are consistent with the greater overall complexity of our conversation domain compared to past systems, as discussed in Section 2. Furthermore, it’s possible that, because of the knowledge that the expert annotators had of the “intended” responses, non-expert users of the system may be more tolerant of turns that expert annotators regard as infelicities. Nevertheless, we view these expert annotations as a useful benchmark for future improvement in this domain.

In the crowdsourced evaluation, we disaggregated the “response appropriateness” annotation into four separate criteria; a deficiency in any one of these criteria may in principle lead a response to be judged as inappropriate. From the crowdsourced ratings summarized in Table 2, we observe that the Eta-based SOPHIE framework was rated relatively highly in fluency/naturalness, consistency with the role of a patient, and appropriateness of emotions. Furthermore, the SOPHIE system notably outperformed the fine-tuned DialoGPT model in these metrics. We take these results to be an indication that the customizability of Eta provides a distinct advantage over an end-to-end deep learning approach, since our system allows the interpretive rules and responses to be designed with a particular role and character in mind. In contrast, the fine-tuned DialoGPT model would sometimes confuse its role in the dialogue.
(e.g., act as a doctor rather than a patient), an issue related to the tendencies of end-to-end neural dialogue models to hallucinate false or contradictory knowledge [15].

The lowest ratings for both sets of responses – and the smallest gains by Eta above the neural baseline – were concerning whether the agent demonstrated understanding of the previous turns. This illustrates that the superficial pattern-matching rules used by Eta are not, at least in their current form, sufficient to demonstrate understanding. Interestingly, the distribution of ratings for Eta on this question in Figure 4 is strongly bimodal, unlike DialoGPT. We believe that this pattern attests to the results found in the expert annotations, namely, that the conditional probability of Eta generating a good response given that it extracted a gist clause is high, but when the system fails to extract a gist clause, it generally makes clarification requests or other inappropriate responses.

Both the expert correctness evaluations in Table 1 and the crowdsourced quality evaluations in Table 2 indicate that the low accuracy of gist clause interpretation is the primary limitation of our approach, although this step allows for the generation of high-quality responses when successful. While this problem can be mitigated to some extent using well-designed fallback response generation strategies, it is a major obstacle towards the practical application of our current system for clinical training. The usability of our system would be greatly improved by incorporating higher coverage mechanisms for gist clause extraction, such as those discussed in Section 5.

### 4.6 Error Analysis

In order to understand the future steps necessary to improve gist clause interpretation and response generation, we performed a qualitative error analysis of these mechanisms by examining common failure cases in the pilot transcripts and assessing which aspects of the system lead to these failures.

In Figure 5, we provide some “good” dialogue excerpts from the SOPHIE pilot study that were highly rated by annotators, as well as some “bad” dialogue excerpts that were poorly rated and which demonstrate limitations of our current system. Minor ASR errors have been corrected in these examples for readability.

#### 4.6.1 Gist Clause Interpretation

The first class of errors that we observed in interpretation stemmed from ASR errors, which occurred in about 20% of input utterances – as mentioned previously, these primarily were either transcription errors or truncated turns. However, a more subtle issue led to additional downstream errors: the specific ASR tool used for the pilot experiment frequently omitted punctuation in inputs, preventing our system from properly chunking sentences or distinguishing questions from statements. For example, in one scenario, the ASR processed the user’s response to SOPHIE as “Imagine this is really scary to hear that the treatment did not work what is your understanding right now of where we’re at”, from which SOPHIE extracted the gist clause “the test results show that the radiation is not working.”. Although the user’s input included both a statement and a question, the lack of punctuation in the input caused the system to miss the question. Errors within this class can likely be solved by using a more accurate ASR tool.

A substantial second class of errors included cases where the correct pattern transduction rules were present, but a weaker match based on superficial cues preempted the correct match. Since pattern matching proceeds in a cascading manner, as discussed in Section 3.4, only the first match for a particular clause will be used. For example, the input “Do you want me to call your daughter later to talk about our visit today” was matched to a very generic pattern for the gist clause “Do you have any children?”, preempting the proper match for “Do you want me to contact a family member?”. One possible solution to this issue would be to instead retrieve, score, and rank all possible matches, at the expense of some amount of efficiency.

A final class of errors were scenarios in which the user inputs involved either out-of-domain topics or unrecognized words. For example, several users used the term “long mass” to refer to SOPHIE’s cancer – a term which was not included in SOPHIE’s lexical features. These cases would cause SOPHIE to fail to extract a gist clause. Augmenting interpretation with a “soft” fallback matching strategy in the case of out-of-domain inputs would likely help mitigate these errors.

#### 4.6.2 Response Generation

The vast majority of failures in response generation were linked to the failures in gist clause extraction described above. In such cases, we employed two types fallback strategies for response generation in the SOPHIE agent: The system may either make a generic clarification request, or the system may give a default response (dependent on the particular schema) that reacts generically to the user and pushes the conversation forward. Examples from the SOPHIE pilot for each of these two strategies are shown in Figure 5.

As can be seen in these examples, both fallback strategies have advantages in some scenarios, but fail in other scenarios. For instance, when users were asked to clarify an utterance, they would sometimes simply repeat their previous utterance, resulting in another failure to extract a gist clause. Moreover, a clarification request after a lengthy response would cause frustration for the user. On the other hand, generating a generic default reaction may be inappropriate if the user asked a question, mentioned important information, or expressed empathy. Thus, neither strategy for response generation is sufficient for accounting for errors in the interpretation pipeline.

Finally, in a fraction of cases where a correct gist clause was extracted, SOPHIE would occasionally generate a response which was either redundant or contradictory given the broader conversational context, due to a lack of commonsense reasoning capabilities in the current system.

### 5 Limitations and Future Work

In addition to the limitations of our system analyzed in Section 4.6, we also acknowledge limitations of our evaluation method. First, the sample size of 15 transcripts that we obtained from the SOPHIE pilot experiment is relatively small, making it potentially difficult to interpret our results.
SOPHIE: I’ve already scanned the results, but don’t understand what they are telling me. Could you explain what they mean?
User: So unfortunately Sophie I have some bad news. It looks like the cancer has grown and spread.
SOPHIE: Those are not the words I wanted to hear. I mean, I was bracing for the worst, since I could tell by the pain that it’s bad. But to hear that the cancer has spread is quite depressing. What does it all mean for me?
User: How much information would you like to know about the prognosis?
SOPHIE: I feel very anxious about my condition. I feel like it’s gotten worse. I want to know what this means for me.
User: What concerns do you have about the future?
SOPHIE: What worries me the most is just how uncertain my future is. Some days I believe my condition will get better and others I brace myself for the worst. Is there any way you can let me know what I should expect?

(a) A “good” excerpt in which the doctor informs SOPHIE that her cancer has spread, and discusses her concerns for her future.

SOPHIE: I still don’t understand how this happened! Just three months ago I didn’t even know I had cancer.
User: I agree, lung cancer in particular is a very difficult diagnosis for patients and their families.
SOPHIE: Cancer is such a challenging disease to live with. But I appreciate all you’ve been doing to help.
User: I haven’t told my family everything yet. I wanted to wait to talk to you first. What should I say to them?
User: How much information do you want them to know?
SOPHIE: You know about my cancer already, but they don’t know how bad it is. How should I discuss these with them?
User: Being honest is going to help the most, and if you’d like they can come to visit and we can help you talk to them.
SOPHIE: Okay. It will be hard to give them such bleak news, but they deserve to know the truth.

(b) A “good” excerpt in which the doctor gives SOPHIE tips on how to break the news to her family.

SOPHIE: I know it’s not easy to say for sure. I would like to be fully prepared for the future though. Is there really no way to tell what my tests mean for me?
User: Absolutely. I think that what I have to say might be a little difficult to hear, but it seems like you want to know. Based on the test, it looks like the cancer really has not responded to the treatment, and that there may not be any further treatment available to shrink the tumor.
SOPHIE: I am sorry, I’m having trouble understanding. Could you try saying it again, a bit more clearly?
User: Sure. I was just saying that, as you suspected from the scan, it looks like the radiation treatment was not entirely successful, and that the tumor size has not shrunk.
SOPHIE: I’m not surprised to hear the radiation did not help, as my pain has gotten worse in these past couple of weeks. But what does this mean for my future? How long do I have left?

(c) A “bad” excerpt in which the system fails to extract a gist clause from the user’s input, causing it to employ a clarification request fallback so that the user rephrases their utterance in a way that the system understands.

SOPHIE: I’m still not sure how I should break this news to my family. I don’t want them to be worried.
User: Would it help to have some more information about what the scan showed?
SOPHIE: Sorry, I didn’t catch that. Could you repeat that one more time using a different phrasing?
User: Would you like to talk about the results of the ct scan?
SOPHIE: I’m not hearing you very well. Would you mind speaking more clearly?
User: Would you like to talk about the results of the ct scan?
SOPHIE: Sorry, I still didn’t quite understand. I have some more questions, maybe we can come back to it.

(d) A “bad” excerpt in which the system generates multiple clarification requests, causing the user to simply repeat the utterance that lead to a gist clause failure and significantly disrupting the conversation.

SOPHIE: I’m just scared about it getting worse, and not being able to be there for my family.
User: I hear that family is very important to you.
SOPHIE: I suppose these things can be hard to predict.

(e) A “bad” excerpt in which the system generates a default generic reaction due to a failure to extract a gist clause. However, the particular default reaction given in this case is an inappropriate response to the empathetic behavior from the user.

Fig. 5

due to the variability in the abilities of the users. Second, the annotations collected only assess local coherence, i.e., whether a response is appropriate given the immediately preceding turn, and don’t measure global coherence across the whole dialogue. In the future, we aim to collect additional transcripts from a scaled-up clinical trial, with additional screening of participants.

Despite the relatively high error rate of our system, [42] observed that users of our system in the SOPHIE pilot experiment were still able to meaningfully interact with the virtual human to improve their patient communication skills. We believe that even in cases where the system doesn’t understand the user well, the system’s production of emotionally and thematically appropriate responses provides the user with opportunities to practice empathetic behavior and other skills.

One direction of future work is to improve the system’s understanding through the development of less brittle, more general interpretive methods. In ongoing work, we’re experimenting with the incorporation of robust neural methods of gist clause extraction, such as few-shot prompting of large language models; past work has demonstrated that similar techniques are effective at mapping user inputs to more constrained forms [43].
Another direction of future work is to incorporate mechanisms for response generation that allow the system to generate fluent and appropriate responses even when it fails to extract an explicit gist clause from input. To this end, we are exploring an approach that uses a large language model to generate responses, conditioned on relevant schema knowledge and retrieved episodic memories of the agent using the extracted gist clause (if any).

Finally, we aim to enable deeper semantic understanding through the incorporation of a broad-coverage neural English-to-ULF parser – such as those explored in [44], [45] – as well as to explore the integration of commonsense knowledge and more thorough affect models, such as those explored in [30], [31], into our system. The conjunction of these would allow SOPHIE-like systems to use reasoning to improve global coherence and generate more relevant affective responses.

6 Conclusion

In this paper, we described a flexible schema-guided dialogue framework that is capable of producing fluent, emotionally appropriate, and thematically appropriate dialogue for the SOPHIE virtual standardized cancer patient, allowing medical practitioners to practice important patient communication skills. Furthermore, our dialogue management framework is general and can be expanded to other types of affective dialogues through the creation of new sets of schemas and pattern transduction trees. Future work will aim to explore methods for improving coverage and robustness of our semantic interpretation component, as well as better fallback strategies for response generation, which will improve the overall ability of the system to understand the user and generate appropriate responses.

Acknowledgments

This work was supported by DARPA grant W911NF19-1-0029, NSF Award IIS-1750380, and NSF Award #1748969.

References


Ehsan Hoque received his Ph.D. degree from the Massachusetts Institute of Technology in 2013. He is an associate professor of computer science with the University of Rochester where he co-leads the ROC HCI Group. Hoque’s research aims to use techniques from artificial intelligence to amplify human ability. His research has been recognized with the MIT TR35 Award, NSF CAREER Award, Emerging Leader by the National Academy of Medicine, among others. He is a distinguished member of the ACM, a senior member of AAAI, and a member of IEEE.

Benjamin Kane is a fifth-year Computer Science Ph.D. student studying natural language understanding and dialogue technology at the University of Rochester. He received his bachelor’s degrees in Computer Science and Economics from the University of Rochester in 2019. His current interests include the development of general-purpose dialogue frameworks that can be used to enable collaborative and open-ended communication between humans and virtual agents, as well as modeling beliefs, intentions, and reasoning in dialogue.

Catherine Giugno is an undergraduate researcher at the University of Rochester, pursuing her bachelor’s degrees in Computer Science and English. Her research interests are in the utilization of creative writing and story development for the creation of realistic conversational AI systems.

Lenhart Schubert pursues research in language, dialogue agents, knowledge representation, and reasoning, relevant to agents with common sense, self-motivation, and the ability to acquire knowledge through language. He received a Ph.D. from the University of Toronto in 1970, was a postdoctoral fellow at Johns Hopkins University, a faculty member in Computing Science at the University of Alberta from 1973-1988, and has been a professor of computer science at the University of Rochester, since 1988. He has over 150 publications, was an Alexander von Humboldt Fellow and is a AAAI Fellow.

Kurtis Haut received his bachelor’s degree in 2018 in Computer Science and Business from the University of Rochester. He began his PhD work during the Summer of 2019 and was awarded a Masters in Computer Science from the University of Rochester Hajim School of Engineering in Spring 2021. His research interests include understanding human behavior by applying artificial intelligence and machine learning techniques as well as human computer interaction broadly. Past research includes deceptive communication, competitive debate, using deepfakes to measure bias, developing assistive technology for PD patients and end of life communication.

Caleb Wohrn Caleb Wohrn was an undergraduate researcher in the University of Rochester Human Computer Interaction Lab, where he worked on projects aimed at addressing social issues including racial justice, climate change, and healthcare. His primary research interests are natural language processing and data analytics. After receiving a B.S. in Computer Science with Honors in Research from UR in 2022, Caleb began work as an Implementation Consultant at Fast Enterprises, developing solutions for the California Department of Tax and Fee Administration.