Advances in Cognitive Systems X (20XX) 1-6 Submitted X/20XX; published X/20XX

Managing Casual Spoken Dialogue Using Flexible Schemas, Pattern Transduction Trees, and Gist Clauses

Seyedeh Zahra Razavi SRAZAVI@CS.ROCHESTER.EDU
Lenhart K. Schubert SCHUBERT@CS.ROCHESTER.EDU
Mohammad Rafayet Ali MALI7@CS.ROCHESTER.EDU
Mohammed Ehsan Hoque MEHOQUE@CS.ROCHESTER.EDU
University of Rochester, Rochester, NY 14627 USA

Abstract

We describe an approach to managing casual spoken dialogues in a virtual human dialogue partner, designed to enable the user to practice basic dialogue skills. We use script-like schemas to loosely guide the dialogue, where the steps of a schema are specified in a declarative language that allows for both explicit and abstract description of speech acts. The steps become particularized and can be added to or omitted as the dialogue proceeds. Internally, the dialogue manager casts its own contributions and those of the user as *gist clauses* – simple, explicit, context-independent versions of what was actually said. Both gist clause computation, and generation of responses, is enabled by hierarchical pattern transduction. These methods show promise in the implementations we have so far developed.

1. Introduction

Engaging in casual, but meaningful conversation ranging over multiple topics is a human skill that still largely eludes AI systems. Recent spoken dialog systems with a meaningful goal include systems designed to help people improve their social skills (Tanaka et al., 2015; Ali et al., 2015) or perform better in job interviews (Anderson et al., 2013; Hoque et al., 2013), to train ASD children with communication deficits through collaborative tasks (Bernardini et al., 2014; Tartaro & Cassell, 2008), and to identify psychological distress indicators in veterans (Rizzo et al., 2011). But most of these conversational agents lack a module for natural language understanding, and dialogue behavior based on such understanding. With no significant comprehension of the user's inputs, they are unable to generate comments that respond meaningfully to those inputs; instead they focus on prosodic features and nonverbal behavior, providing feedback on that basis.

Spoken language understanding in the context of topically broad dialogue has proved to be very challenging, for multiple reasons:

- State-of-the-art speech recognition systems are still far from perfect. Errors in the output can make it almost impossible to apply standard syntactic parsing and interpretive techniques, and can lead to radical misunderstanding.

- Casual spoken language, even when accurately transcribed, contains deviations from strict grammar, meta-comments, repeated words, filler sounds, broken-off sentences and restarts, etc., greatly limiting the applicability of text understanding methods.
- Traditional dialogue management methods are typically either flowchart-driven or dialogue-state-driven, iteratively mapping the current state and type of input to a new state and possible output. But such methods are not well-suited to open-ended conversations, which may range over multiple topical threads and may encounter unexpected inputs from the user.

Our approach to dialogue management is based on the hypothesis that human cognition and behavior rely to a great extent on dynamically modifiable schemas and on hierarchical pattern recognition/transduction. Thus we propose a dialogue manager that follows a flexible, modifiable schema – in effect a plan of interaction with the user, subject to change as the interaction proceeds. At any time, the most recent question asked by the virtual agent provides the context in which the user's input is interpreted. The dialogue system uses hierarchical pattern transduction to map the user's input to one or more explicit context-independent English "gist-clauses". This approach to dialogue management has been implemented in the LISSA virtual agent (Ali et al., 2015), which is intended to help people improve their social communication skills. LISSA provides continuous feedback on the user's prosody and nonverbal behavior (gaze, smiling, etc.) as well as end-of-session summary feedback. The agent has been used so far in two different studies. In a speed-dating study, evaluation of the transcripts from the automatic system showed no difference from those using a Wizard-of-Oz setting, as judged by research assistants (Ali et al., 2015). Another exploratory study involved seven teenagers with autism spectrum disorder (ASD), and the users not only found the conversation with the virtual agent natural but also expressed interest in further interaction with such a system (Razavi et al., 2016). Further studies are under way both to achieve more precise evaluations, and to broaden the range of applications. The most exciting aspect of the LISSA project is its potential for ubiquitous use by individuals with social interaction difficulties, allowing them to practice skills repeatedly, privately and without fear of stigma. Our dialogue management techniques seem well-suited for systems of this type.

In the remainder of this paper, we first describe some recent virtual agents that use spoken language. In section 3 we present the high-level structure of our dialogue manager, illustrate its capabilities with an extended excerpt from a conversation with LISSA, and explain the use of schemas, pattern transduction, and gist clauses in context-dependent interpretation of user inputs, and generation of outputs. In section 4 we explain our novel approach to input interpretation and output generation more fully. In section 5 we describe our dual implementations for two user groups (university students and ASD teens) and describe the quite encouraging results of our preliminary experiments. In section 6 we summarize our conclusions and propose potential future research lines for improving and extending the system.

2. Related Work

Research on conversational virtual agents has expanded rapidly in recent years. The improvements in speech recognizer quality, along with more effective techniques for face-to-face interaction between humans and computers has convinced researchers to use spoken virtual agents in various applications. For example, TARDIS (Anderson et al., 2013) is a platform for practicing a job interview in which users' emotions, moods and attitudes are evaluated based on head gaze, voice activity, gestures and postures. MACH (Hoque et al., 2013) is a rather similar framework in which the system gives continuous feedback on users' smiles, head movement, intonation, etc., throughout a simulated interview. LISSA (Ali et al., 2015) showed that young people who practice their communication skills with a virtual agent are more likely to improve in a speed dating task compared to those who watched videos on social skills improvement . In SimSensei (DeVault et al., 2014) the virtual agent leads an interview with the user with the goal of detecting psychological distress indicators. An earlier system developed by the same team, SimCoach (Rizzo et al., 2011), was used to detect signs of PTSD in military personnel. One of the earliest "companionable" systems, HWYD (Cavazza et al., 2010; Pulman et al., 2010), was designed for conversing about the user's workday while providing appropriate emotional feedback based on features of the user's speech.

Other research lines focused on applications of conversational virtual agents in helping people with special needs. (Yaghoubzadeh et al., 2013) explored the feasibility of using spoken-language interaction in a virtual assistant to elderly people or people with cognitive impairments. Other research investigated the effectiveness of interaction with computers in children with ASD. ECHOES (Bernardini et al., 2014) is an interactive learning environment designed to improve communication skills of children with ASD by involving them in a collaborative task with an embodied virtual agent. (Tartaro & Cassell, 2008) showed that children with ASD interact better when they are involved in a collaborative story telling task with a virtual agent rather than with a human peer.

Although interaction with conversational virtual agents seems to be effective in various applications, most of the state-of-the-art virtual agents are weak in terms of language understanding. Many systems either do not support any language understanding, such as MACH (Hoque et al., 2013) and ECHOES (Bernardini et al., 2014), or use Wizard-of-Oz technique to lead a meaningful conversation with the user, such as the initial version of LISSA (Ali et al., 2015) and (Tartaro & Cassell, 2008). Among those with a distinguishable dialogue manager, different methods are used. In HWYD hand-written patterns are matched against the input, after labeling with named entities. The extracted information is passed to the dialogue manager and new conversational goals are derived from unfilled slots. Based on four classifier outputs and the avatar's current question, the dialogue manager chooses from 100 fixed utterances. In SimCoach (Rizzo et al., 2011), the FloReS dialogue manager (Morbini et al., 2014) dynamically chose conversational moves based on several hundred subdialogue networks ("operators") so as to optimize expected future rewards. Scaling up this method proved problematic because of the need to define large numbers of rewards to ensure effective choices of conversational moves. Thus, in the more recent SimSensei system by the same team, the dialogue manager is much simpler, basing its output choices on the features determined by four input utterance classifiers: the generic dialogue act type; positive, negative or neutral sentiment; domain-specific anticipated keywords (e.g., name of a city as response to a particular avatar question); and domain-specific dialogue acts (e.g., various ways the user might respond positively to a particular avatar question).

Some other systems treat the turns in a dialogue as transitions in a finite state machine (Matsuyama et al., 2016; Zhao & Eskenazi, 2016; Lee & Stent, 2016) and try to find the best dialogue policy using an optimization technique. This method is viable if it is possible to identify a fixed

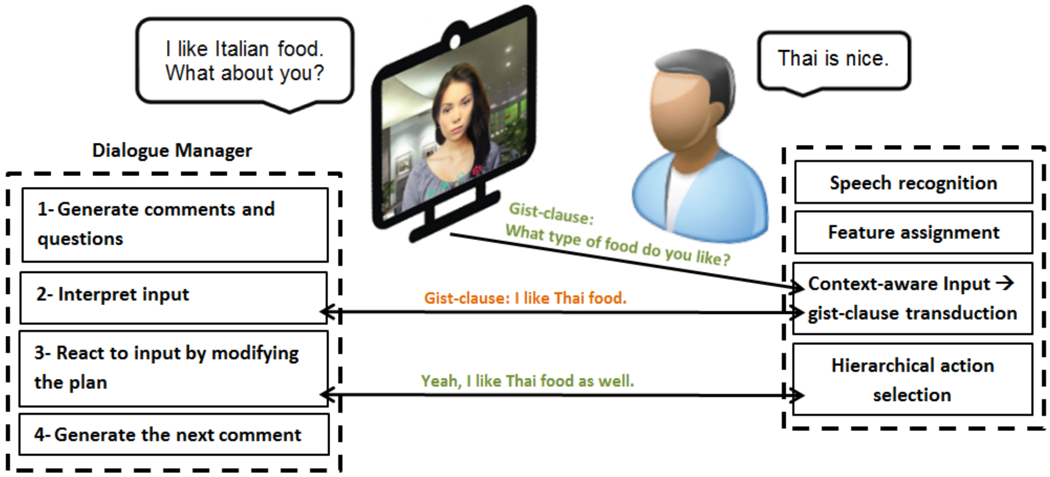

Figure 1. Overview of Dialogue Manager.

number of conversational states (e.g., in certain domain-specific dialogue systems), but is difficult to extend to open-ended conversational agents. Some task-oriented spoken dialogue systems, such as TRIPS (Ferguson et al., 1998), allow for sophisticated collaborative planning, recognition of user intentions, and domain problem solving, but it is an open question whether these techniques can be extended to broader dialogues involving human concerns in daily life and personal relationsips.

3. Overview of the Dialogue Manager

As diagrammed in Figure 1, our virtual agent, LISSA, leads a conversation primarily by asking questions of the user, often with some prior comments about herself. Following each user's response, LISSA might show one of the following behaviors:

- Making a relevant comment on the user's input;
- Responding to a question, if the user asked one (typically, at the end of the input);
- instantiating a subdialogue, if the user switched to an "off-track" topic (an unexpected question may have this effect as well.)

3.1 Sample Dialogue

Figure 2 shows the central portion of a conversation between a user and LISSA. The example is from the "speed-dating" version of LISSA, allowing us to show that LISSA can extract essential information even from relatively wordy inputs, and can respond to user questions. (The ASD-teen dialogues were typically less wordy and lacked questions from the users.) The excerpt is not particularly unusual, except perhaps that the speech recognizer functioned rather accurately for this particular user. However, LISSA handles less accurate SR output quite robustly.

LISSA: And what do you not like about Rochester?
User: I don't like the weather really. It's a bit cold.
LISSA: Yeah, I find it cold here too. Is there anything you would change?
User: I guess the one thing I would change is the weather. Make it a bit warmer.
LISSA: Not much we can do about it.
If you can't tell I haven't seen much of the city at all. What would we do if you took me on a tour?
User: There are a lot of landmarks around here. I would definitely take you to the Bar District show you some local eateries because there is not much to do here beside that sort of stuff. I guess the coolest stuff is to take you outside of the city and show you some natural area: trail, park, etc.
LISSA: Oh, I always love spending time outdoors.
One thing I always wonder about are good places to eat. You might not think it by looking at me but I think restaurants are just great. I love to watch people enjoy their food. I also love the way that every place has its own unique atmosphere. It doesn't have to be fancy. I also love those dirty spoon holes in the wall restaurants that have a really fun and cool vibe. Could you tell me about your favorite place to eat here in Rochester?
User: I guess I feel like that sort of thing. My favorite place to eat would have to be the place called Flavors of Asia. A lot of Thai food, fusion cuisine really good stuff and... you would like it.
LISSA: That's great. I like Thai food.
And could you tell me about the Rochester delicacy? What's this whole garbage plate thing about?
User: Well it is kind of than the name a plate of garbage you're eating the nastiest and the greasiest food but when it's all put together it tastes very good. Excellent to eat when you're on the outside going out after a long night I would like to consider a delicacy more like an experience.
LISSA: Sounds like it's high-calorie but good.
By the way, have you been to Dinosaur barbecue?
User: I have been I've only been once though. Have you?
LISSA: Well, just in my imagination. Everyone seems to like it.

Figure 2. Sample dialogue between a user and LISSA

3.2 Schemas and Pattern Transduction

As already mentioned, LISSA follows a *dialogue schema* – a list of expected successive events in a dialogue. These events are instantiated in the course of the conversation. The steps of the schema may specify particular utterances, but also abstract ones to the effect "The user answers my question", or "I react to the user's answer". The steps have variable names, and these are dynamically instantiated with constant values in the course of the dialogue, which then serve as names of both the utterance event and its propositional content.

Our concept of a (dynamic) schema has some similarities with so-called Schankian scripts (Schank & Abelson, 1975), though our schemas are intended for *guiding* behavior, not only for

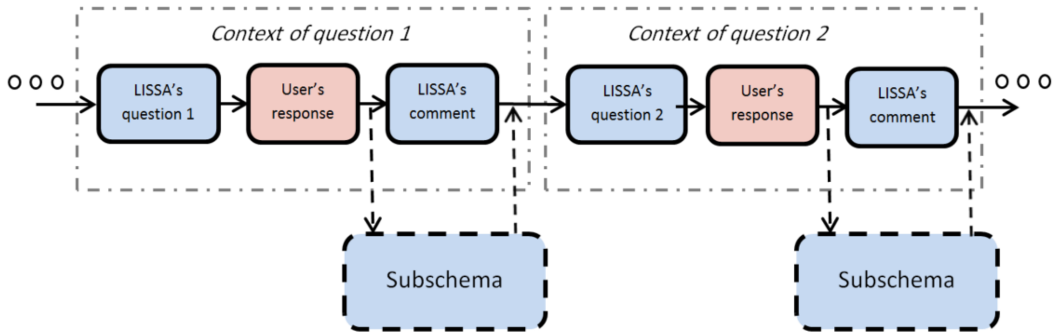

Figure 3. Dialogue acts performed in a LISSA conversation with a user

```
?a1. (me say-to.v you             '(Hi\, I am Lissa\. What is your major?))          ; asking LISSA's question
?a2. (you reply-to.v ?a1.)                                                     ; getting the user's response
?a3. (me react-to.v ?a2.)                                                      ; LISSA producing a comment
                              (for example "So, we are both scientifically minded." as the reaction)
```

Figure 4. A segment of a schema consisting of three LISSA actions

interpreting or predicting it. As such, schemas can function as plans, while allowing for actions by other agents and for exogenous events. In our general conception, schemas allow specification of participants and static and temporal constraints, as well as conditionals and iteration. But for present purposes a schema just consists of a predicative header (for the type of action or event, allowing argument variables) and a sequence of variables corresponding to system and user actions, along with corresponding action specifications. The actions are expected to occur in the given order by default, and thereby guide LISSA's actions and help interpret user inputs.

The typical sequencing of steps (dialogue acts) in LISSA's main schema is diagrammed in Figure 3, and an example of a schema segment is shown in Figure 4. As seen in the figures (and implied in the preceding discussion), the specified events typically come in threes, where the first specifies an explicit LISSA output, usually ending in a question, the second states that the user replies to that output, and the third states that LISSA reacts to the user's reply.

Throughout the conversation, the replies of the user are transduced into simple, but explicit, English sentences – gist clauses. The interpretive process begins with automatic assignment of syntactic or semantic features to many of the words in the input. (In section 4 we will discuss how features enable pattern matching without exclusive reliance on specific words.) The dialogue manager then selects pattern transduction hierarchies relevant to interpreting the user's response to LISSA's preceding output, and applies the rules in the selected hierarchies to derive one or more gist clauses from the user's input. The choice of the pattern transduction hierarchies for interpreting an input is itself based a pattern match to a gist clause representation of LISSA's question. In part, this is done for robustness and portability of transduction methods (we comment further later on).

Next, the system applies hierarchical pattern transduction methods to the gist clauses derived from the user's input to generate a specific verbal reaction, or to segue to another transduction tree (e.g., to react to an off-topic input), or to choose a subplan to be inserted into the current plan. Subplans are particularly useful for generating reactions to multiple aspects of the user's input, and may in principle lead to the insertion of further subplans. LISSA's verbal reactions are usually comments on the user's answer to LISSA's question, but as mentioned above can also be (or include) an answer to a question posed by the user. In any case, the plan will be modified and LISSA will produce the relevant output utterance(s). Then the plan continues with the next action in the main schema.

As an example, suppose that as in the middle of Figure 2, LISSA asks the user about the user's favorite place to eat. Assume that as in Figure 4, the variable instantiated for that action is ?a1. Then the gist clause version of LISSA's question associated with ?a1 will be, "What is your favorite restaurant?". The next step takes this as a context for interpreting the user's input (?a2) as a gist clause, for instance, "My favorite restaurant is a Thai restaurant". This gist clause is used in turn to generate a sympathetic reaction (?a3) such as : "That's great! I like Thai food". LISSA then follows the rest of the plan by proceeding to the next output specification.

The action variables and their contents are progressively instantiated, and in this way provide a record of past, current, and expected events. However, the expected steps can in principle be deleted, replaced, or intercalated with steps from other schemas or subschemas. A distinctive characteristic of the schema language is that steps are formulated logically. As seen in Figure 3, the step syntax allows both for quoted terms (enabling statements to the effect that the system utters a particular phrase or sentence), and for event or propositional arguments, including reference to previously instantiated action variables. While we are currently not allowing for inference in a significant way, the schema language is intended ultimately to interface smoothly with an inference engine drawing on propositional knowledge about the world and about the current context. Thus responses could be based on inferences, though we have found our hierarchical pattern transduction methods quite adequate as "inference methods" for responding to inputs to the LISSA variants built so far.

4. Context-dependent language understanding

We now provide some further details of input interpretation in the context of a LISSA question, and the choice of a reaction to the interpreted input.

4.1 Conceptual Pattern Matching via Features

The features used to annotate input words specify classes of words that are syntactically or semantically similar in some way, such as synonyms (e.g., friend, pal, buddy), syntactic classes (e.g., anaphoric pronouns or wh-words), or semantic classes (e.g., challenging academic subjects, or liking-verbs). We use both general features, attached in all contexts, and context-specific ones, such as those relevant to subjects potentially studied by a college student, or the types of local restaurants. Features are recursively added to words, i.e., if a feature is added, so are the features of that feature, etc. For example, 'astrophysics' and 'Austin' might be expanded respectively into the following:

(astrophysics physics challenging-course academic-course),
(Austin Texas Southern-Cities Southern-States US-Cities US-States).

So in writing match rules we can use features at various levels of abstraction, and this supports multi-level pattern matching, where patterns at successively deeper levels typically test for ever more specific properties of the annotated input. Since various types of user responses are possible for a given question, there will generally be alternative patterns at any given level. Once a match is found, more specific patterns will be tried at the next, deeper level, until such a pattern is found that is paired with a result template or a "directive". Result templates typically contain both predetermined words and words obtained from the input through binding of match variables. In interpretation trees, templates specify the structure of gist clauses based on the matched input; in trees for reacting to an answer, templates specify the form of an output. Directives may redirect a pattern match to another transduction tree (e.g., for unexpected inputs, such as complimentary or uncomplimentary ones), or in the case of reactions to an input they may specify that a subplan should be inserted into the dialogue plan (potentially with parameters supplied by the pattern match).

4.2 LISSA Questions as Context for Interpreting User Responses

The interpretive trees selected for interpreting answers to a LISSA question may generate multiple gist clauses representing the user's input. As we will indicate, LISSA's reaction might then cover multiple aspects of the user's input, via insertion of an utterance subplan into the main plan.

The root of a transduction tree specifies its name while the interior rule nodes specify patterns at various depths, and as was seen above, leaves specify output templates or directives. The search algorithm is depth-first, looking through patterns at a given level for a match to the input. Figure 5 shows a transduction tree for extracting gist clauses from the user's input. At the top level, a subtree is selected based on the most recent LISSA question, then different subtrees are matched against the input to extract all possible gist clauses (further details below).

Interpreting the user's response in the context of the question asked is important since user responses often convey little meaning out of context. For instance, the reply "Yes, I have." to LISSA's output "I like the new Star Wars movie; have you seen it?" has the actual meaning "I have seen the new Star Wars movie," while the same reply to the input "I like the Dinosaur Barbecue; have you been there?" means "I have been to the Dinosaur Barbecue." By using the gist clause representation of LISSA's question (e.g., "Have you seen the new Star Wars movie?") as the context for selecting an interpretation tree, the dialogue manager is able to match appropriate patterns to the user's reply, and thus find a gist clause interpretation of that reply (e.g., "I have seen the new Star Wars movie"). Every (you reply-to.v ...) action encountered in the dialogue plan is interpreted in that way.

A major advantage of internally rendering the user's answers as explicit, more or less context-independent gist clauses is that this makes many transduction trees portable from one dialogue setting to another. For example, a gist clause such as "My favorite food is Thai food" can fairly safely be responded to with, say, "I prefer Japanese food", or "Mine too, especially Massaman

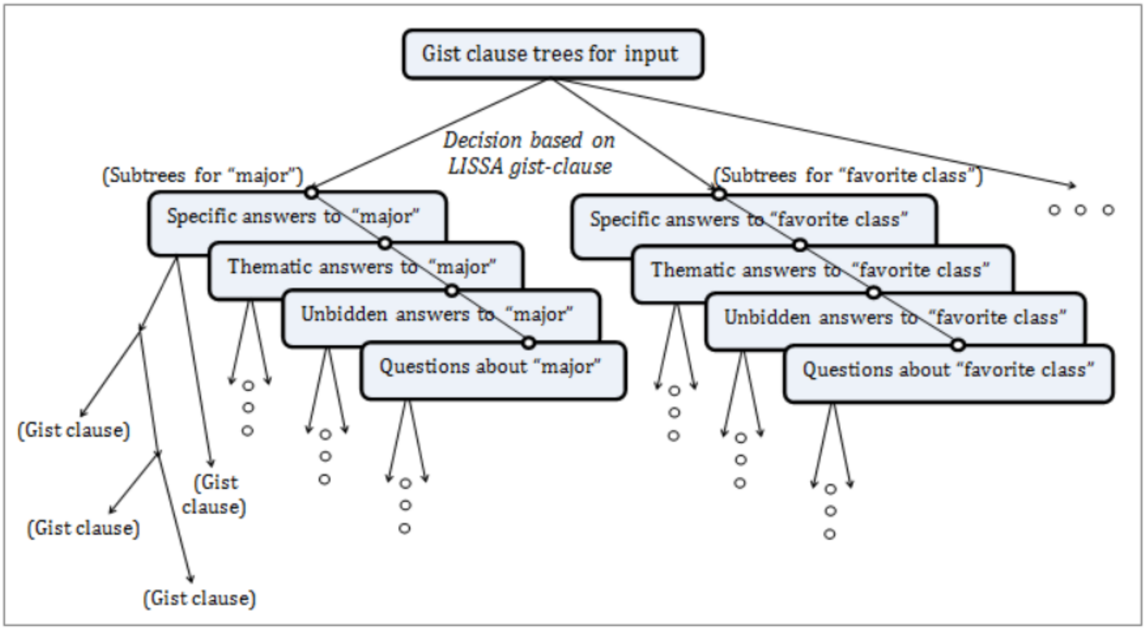

Figure 5. A transduction tree to extract a gist clauses from the input.

curry", etc. – no matter who the dialogue partner may be or what other topics may be touched on in the dialogue.

In general, four interpretation trees are applied to each user response, containing the following sorts of rules:

- Rules for specific answers: These recognize answers that directly respond to the LISSA question.
- Rules for unbidden answers: These look for extra information that may answer questions LISSA intended to ask later. For instance, LISSA may ask about the user's favorite eatery, and may intend to ask subsequently whether the user had ever been at the Dinosaur Barbecue. But LISSA will delete the latter question from the dialogue plan if the user's favorite eatery turns out to be the Dinosaur Barbecue.
- Rules for thematic answers: These detect general themes in an answer when no direct answer can be found. For example, an answer to a question about the user's favorite eatery could be something like, "I like to try various restaurants with various cuisines", and LISSA will try to extract a gist clause from this.
- Rules for questions: These look for possible questions directed to LISSA, typically reciprocal questions. For example, if LISSA asks "What is your favorite food?", the user might reply, "I like Thai. What about you?" Because the user's question is interpreted in the context of LISSA's prior question, it will be appropriately interpreted as asking for LISSA's favorite food.

For longer inputs, pattern matching is done on overlapping 10-word chunks, and thus multiple gist clauses representing the user's input may be generated. After interpretation of the user's input

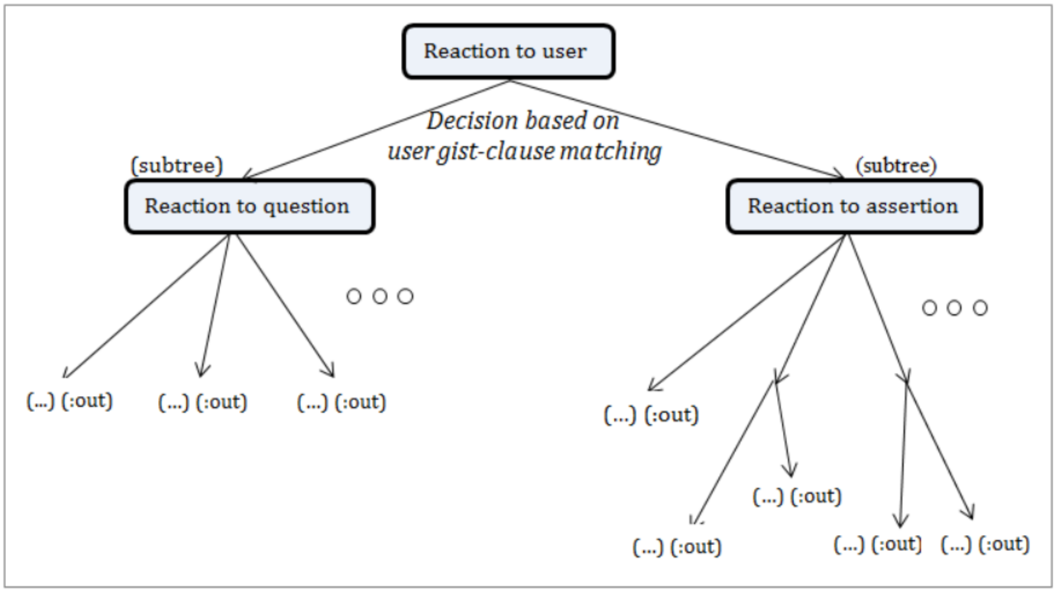

Figure 6. Transduction tree to generate an appropriate LISSA output.

and extraction of all possible gist clauses, another choice tree is used to generate the appropriate LISSA reaction.

For reasons that may not be immediately obvious, we also attach gist clause interpretations to LISSA utterances for which user responses are expected. For example, for LISSA's utterance "I like Italian food. What about you?", as in Figure 1, we attach the gist clause "What type of food do you like?". A transduction tree for interpreting the user's response is then selected by pattern matching against that gist clause. (This pattern match uses the same hierarchical methods as are used for interpretation and response generation.) Compared to direct attachment of a fixed interpretation tree (or set of trees), this method again promotes robustness and portability in building variant applications. For example, if the question about food preference comes up at different points in other dialogues, the designer need not remember the names of the interpretive trees used, but can simply provide a similar gist clause (e.g., "What kind of food do you like best?", "What is your favorite cuisine?", etc.), assuming of course that the hierarchical pattern match against such variants has been designed to map them all to the same interpretive tree(s). Moreover, our design is intended to enable future enhancements that allow inference during the conversation, and reference to previous contributions of the participants; for this we need a record of the semantic content of those contributions, for both participants.

4.3 LISSA's Reactions to User Inputs

Besides the four interpretive trees enumerated above, we construct reaction trees for each LISSA question, aimed at responding appropriately to the user's answer. To perform the action (me react-to.v ...), LISSA accesses the user gist clauses and chooses the correct reaction trees by matching

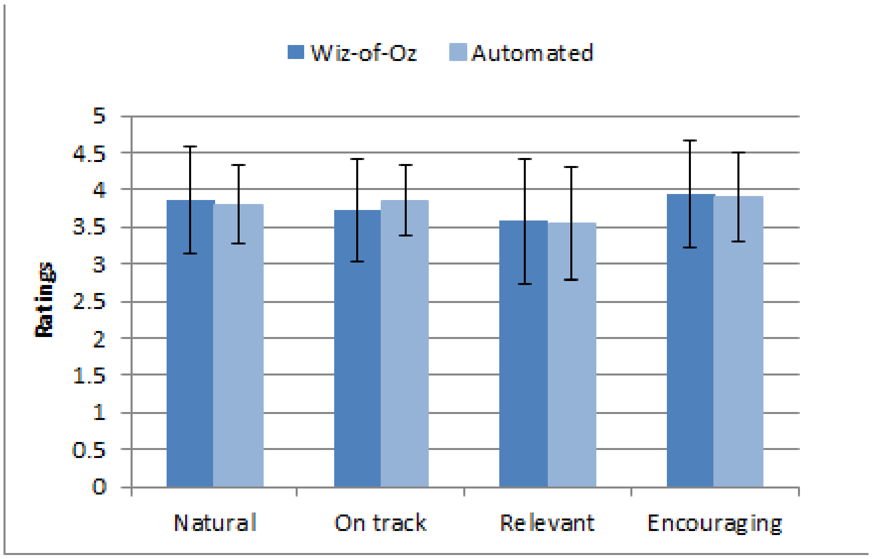

Figure 7. Means and Standard Deviations of Dialogue Ratings in Speed-dating task

them to the possible gist clause forms, and then applies the reaction trees to produce the relevant output. LISSA's reaction is often a comment on the fact, opinion, or preference contained in the user's answer. As in the case of answer interpretation, various alternative patterns may be tried at any level of a transduction tree; also there may be multiple response options (output templates) associated with a matched pattern (different choices may be made in different runs via "latency" parameters). LISSA's reaction might then cover multiple aspects of the user's input, appropriately sequenced, via insertion of an utterance subplan into the main plan. If nothing relevant is found, LISSA will move on to the next step in the schema.

Figure 6 shows a transduction tree designed to generate the appropriate LISSA comment. Relevant subtrees are selected by matching patterns against the user's extracted gist clauses. If the user asked a question from LISSA, the respective subtree will be evoked and provides the appropriate response to the user's question. Otherwise, the hierarchical pattern matching goes deeper into the other subtree and will eventually end up with an appropriate comment as LISSA output.

5. Implementation

The LISSA dialogue manager has been implemented and tested in two potential applications – training for speed dating and training ASD teens in casual dialogues. In order to design the dialogue plan for speed dating we collected the transcripts from a pilot study using a Wizard-of-Oz technique (Ali et al., 2015). Based on the transcripts from two rounds of conversation with 23 local students we designed the transduction trees and implemented the fully automatic dialogue manager.

After developing the automated system, we ran a study with 10 participants and compared the transcripts from the automatic system with those from the human-operated, Wizard-of-Oz prototypes. To evaluate the quality of the dialogue module, we evaluated transcripts from 12 rounds

Table 1. Average rating of post-session survey questions in ASD task

	Average	SD
I think LISSA understood me.	3.8	0.74
My experience of talking to LISSA was almost as real as talking to a real human.	3	1.09
I would use this system to practice my social skills in future.	3.4	0.8

of Wizard-of-Oz interaction, selectively preferring interactions with the least number of manually typed responses (as opposed to responses selected from a menu), and 15 rounds of the automated system. These transcripts were then deidentified, randomized, and distributed to 8 RAs, with each transcript being rated by at least 2 RAs. They rated the videos on four criteria: the naturalness of the dialogue; whether the conversation stayed on track; how relevant LISSA's responses were to the conversation; and the extent to which LISSA's responses encouraged the participants. No guideline was given to RAs; rather they were asked to use their own best judgment. Each criterion was rated on a scale ranging from 0 (not at all) to 5 (completely). Ratings for each transcript were then averaged across the RAs to represent the consensus score for that transcript. The average and one standard deviation of each of the criteria are shown in Figure 7. Further studies on this task are under way to obtain a higher level of confidence in the evaluation of the system.

In the second task, we ran a preliminary study with 7 teenagers (one girl and six boys) ranging in age from 15-17 years, all diagnosed with ASD (Razavi et al., 2016). In this study, each participant had two short conversations with LISSA, and was then asked to evaluate the system by filling a survey; also we had a chat with them about what they did and didn't like about the system. The average results for the surveys filled out by the participants are shown in Table 1. Responses were given on a five-point scale ranging from "strongly disagree" (=1) to "strongly agree" (=5).

Five of the seven participants said that they believed LISSA understood them, and they mentioned this as something they liked about the system (Razavi et al., 2016). The participants also generally affirmed that they would be interested in further use of such a system.

6. Conclusion and Future Work

We have implemented a dialogue manager for a conversational virtual agent, called LISSA. The virtual agent is intended to lead a natural open-ended conversation with the user while providing feedback on the user's nonverbal behavior including smiling, gaze direction, volume and head motion. The dialogue manager follows a flexible, modifiable schema which is updated during the conversation and can be changed as the interaction proceeds. Pattern transduction trees are used to extract the gist clauses from the user's input in the context of LISSA's last utterance. This approach provides robustness in the face of speech recognizer errors, and also provides LISSA with the ability to update the plan and inject subschemas in cases such as users asking a reciprocal question or requesting a repetition. Also we developed a feature assignment technique that allows for abstract as well as word-level pattern matching. The schema structure along with the pattern transduction tree method allow us to extend the dialogues as well as to generate new dialogues with relatively little effort.

There are still substantial challenges in fully developing spoken dialogue systems of the type addressed here, some of which we are beginning to grapple with. One lacuna is a method to recognize the user's end of turn. Current spoken dialogue systems use long pauses (e.g., 2 seconds) as signals for seizing the turn. However, this does not result in a natural conversation. Human turn-taking is known to use many features to detect the turn-yielding points, including prosodic features, lexical, syntactic and semantic features of the utterance (Raux & Eskenazi, 2012; Gravano & Hirschberg, 2011) and even eye-contact (Meena et al., 2014). For the LISSA virtual agent we plan to focus on semantic completion, which has been used relatively little and whose detection is particularly natural in our approach. Instead of processing the entire input from the user we plan to process inputs in small segments and look for points at which a complete gist clause can be formed. When there is a sufficient pause and other relevant speech features are present at such a point, LISSA will be able to take the turn naturally.

Another possible improvement in LISSA is to coach the user to say enough. In some cases, especially in conversations with ASD teens, the users provided responses that were too meager (e.g., "Do you like any of your school subjects?"; "Yeah!"). Our method of context-dependent gist clause extraction lets us detect these inadequate responses. In such cases LISSA can prompt the user with a comment like "Would you like to explain more?" Conversely, users may launch into inappropriately lengthy responses. Again the gist clause extraction method lets LISSA track the response and detect when enough information has been provided to warrant interrupting the user.

Also, we would like to provide users more freedom in steering the conversation. For example, we plan to improve the system's ability to respond to relevant questions posed by the user. One of the most striking differences we observed between ASD teens and non-ASD youths was that the ASD teens never asked LISSA questions. So we plan to motivate users to show interest by inviting them to reciprocate with their own questions.

To summarize, our future lines of work will be aimed at extending LISSA's conversational abilities to enable an even more natural conversational flow. The gist clause method as well as the schema technique should enable the enhancements we listed without adding unmanageable complexity to the system. We are also proceeding to run more studies with actual users in different tasks; the goal is to establish more firmly the effectiveness of our methods from the perspective of users, so that ultimately an agent like LISSA can be used for successful training of users in conversational skills.

Acknowledgements

The work was supported by NSF EAGER grant IIS-1543758 and DARPA CwC subcontract W911NF-15-1-0542. Our thanks to the helpful comments of the anonymous referees. We also gratefully acknowledge the significant contributions made by Chelsea Meyers in the initial work on dialogue management.

References

Ali, M. R., Crasta, D., Jin, L., Baretto, A., Pachter, J., Rogge, R. D., & Hoque, M. E. (2015). LISSA – live interactive social skill assistance. *Affective Computing and Intelligent Interaction (ACII), 2015 Int. Conf. on* (pp. 173–179). IEEE.

Anderson, K., et al. (2013). The TARDIS framework: intelligent virtual agents for social coaching in job interviews. In *Advances in computer entertainment*, 476–491. Springer.

Bernardini, S., Porayska-Pomsta, K., & Smith, T. J. (2014). ECHOES: An intelligent serious game for fostering social communication in children with autism. *Information Sciences*, *264*, 41–60.

Cavazza, M., De La Camara, R. S., & Turunen, M. (2010). How was your day?: a companion ECA. *Proc. of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems: Volume 1* (pp. 1629–1630). Int. Foundation for Autonomous Agents and Multiagent Systems.

DeVault, D., et al. (2014). SimSensei Kiosk: A virtual human interviewer for healthcare decision support. *Proc. of the 2014 Int. Conf. on Autonomous Agents and Multi-Agent Systems* (pp. 1061–1068). Int. Foundation for Autonomous Agents and Multiagent Systems.

Ferguson, G., Allen, J. F., et al. (1998). TRIPS: An integrated intelligent problem-solving assistant. *AAAI/IAAI* (pp. 567–572).

Gravano, A., & Hirschberg, J. (2011). Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, *25*, 601–634.

Hoque, M. E., Courgeon, M., Martin, J.-C., Mutlu, B., & Picard, R. W. (2013). Mach: My automated conversation coach. *Proc. of the 2013 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing* (pp. 697–706). ACM.

Lee, S., & Stent, A. (2016). Task lineages: Dialog state tracking for flexible interaction. *17th Ann. Meet. of the Special Interest Group on Discourse and Dialogue* (p. 11).

Matsuyama, Y., Bhardwaj, A., Zhao, R., Romero, O. J., Akoju, S. A., & Cassell, J. (2016). Socially-aware animated intelligent personal assistant agent. *17th Ann. Meet. of the Special Interest Group on Discourse and Dialogue* (p. 224).

Meena, R., Skantze, G., & Gustafson, J. (2014). Data-driven models for timing feedback responses in a Map Task dialogue system. *Computer Speech & Language*, *28*, 903–922.

Morbini, F., DeVault, D., Sagae, K., Gerten, J., Nazarian, A., & Traum, D. (2014). FLoReS: a forward looking, reward seeking, dialogue manager. In *Natural interaction with robots, knowbots and smartphones*, 313–325. Springer.

Pulman, S., Boye, J., Cavazza, M., Smith, C., & Santos de la Cámara, R. (2010). How was your day? *Proc. of the 2010 Workshop on Companionable Dialogue Systems* (pp. 37–42). Uppsala, Sweden: Assoc. for Computational Linguistics. From `http://www.aclweb.org/anthology/W10-2707`.

Raux, A., & Eskenazi, M. (2012). Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Trans. on Speech and Language Processing (TSLP)*, *9*, 1.

Razavi, S. Z., Ali, M. R., Smith, T. H., Schubert, L. K., & Hoque, M. E. (2016). The LISSA virtual human and ASD teens: An overview of initial experiments. *Int. Conf. on Intelligent Virtual Agents* (pp. 460–463). Springer.

Rizzo, A., et al. (2011). SimCoach: an intelligent virtual human system for providing healthcare information and support. *Int. J. on Disability and Human Development*, *10*, 277–281.

Schank, R. C., & Abelson, R. P. (1975). *Scripts, plans, and knowledge*. Yale University New Haven, CT.

Tanaka, H., Sakti, S., Neubig, G., Toda, T., Negoro, H., Iwasaka, H., & Nakamura, S. (2015). Automated social skills trainer. *Proc. of the 20th Int. Conf. on Intelligent User Interfaces* (pp. 17–27). ACM.

Tartaro, A., & Cassell, J. (2008). Playing with virtual peers: bootstrapping contingent discourse in children with autism. *Proc. of the 8th Int. Conf. on for the Learning Sciences–Volume 2* (pp. 382–389). Int. Soc. of the Learning Sciences.

Yaghoubzadeh, R., Kramer, M., Pitsch, K., & Kopp, S. (2013). Virtual agents as daily assistants for elderly or cognitively impaired people. *Int. Workshop on Intelligent Virtual Agents* (pp. 79–91). Springer.

Zhao, T., & Eskenazi, M. (2016). Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*.