A Mixed Initiative Dialog Framework

Contributors: A. W. Crapo, Alfredo Gabaldon, GE Research
Last revised 7/30/2019

Introduction

The objective of this document is to give a conceptual description of a dialog system in which a mixed initiative conversation between two parties, namely a human user and an artificial intelligence (AI), can occur on multiple topics. Extension to more than two parties is envisioned as possible but not currently in scope. The dialog is envisioned to occur in a simple "chat window" which can be implemented in a text editor. In specific, an Xtext editor is envisioned so that Xtext functionality can be used 1) to constrain the dialog content to that supported by the Dialog grammar to avoid ambiguity (otherwise errors are detected, underlined, and explained with markers in the margin), 2) to enable content assist to provide possible statements or continuations of partial statements, and 3) to enable hyperlinking of all domain concepts appearing in the dialog to their definitions and to additional references in imported models (imported directly or indirectly) and to additional references in this or other dialog file.

Requirements and How to Meet Them

For such a dialog to be supported the following requirements must be met.

1. It must be possible to identify which party has contributed each statement in the dialog.
2. It must be possible to have multiple topics and to differentiate between topics.
3. It must be possible to navigate programmatically between statements in a dialog topic in order to obtain the full sequence of topic statements for analysis by the AI in determining whether to/how to respond.

In addition, the dialog content should be easily understood by a human user. "...cognitive psychologists suggest that three aspects of interface design must be addressed if the interface is to serve as an effective medium: (1) content (what semantic information should be contained in the representation given the goals and tasks of the users, (2) structure (how to design the representation so that the user can extract the needed information), and (3) form (the notation or format of the interface)" (Leveson, 2000).

The first requirement (identifying party "speaking") can be achieved by prefixing all statements made by at least one of the parties with an identifier. In our current two-party dialog between user and AI, we have the AI prefix each statement which it contributes so as to not burden the human user with prefixing his/her statements. It would be perhaps helpful if the system prefixed the human user's statements, or not.

The second requirement can be met in several ways.

1. One approach would be to have a separate editor window for each topic. This seems suboptimal as there would potentially be many windows and the user would not have

enough information available to be able to easily identify the window for a desired topic.

2. Another approach would be to indent all but the first conversation element belonging to a topic. This would result in an appearance similar to how many threaded conversations display/organize their conversation elements. This would permit a tree structure in the conversation on a given topic. One disadvantage is that the user would have to manually provide the correct level of indentation to indicate where in the tree a new element is being added, although "smart" editors will often indent automatically after a newline to the indentation level of the preceding statement. An advantage of this approach is that this would make it easy for a thread to be forked for sub-comments on any statement after another, more temporally recent, follow-on statement has been made.
3. A third option is to separate topics with a blank line. In this case each topic would be seen as a sequence of "flat" conversation elements. There would not be the ability to have threaded sub-topic discussions.

The third requirement is probably not difficult, once an approach to meeting the second requirement is chosen. In any case, from any given statement, it must be possible for the AI to retrieve all parent statements back to the "root" of the topic so as to resolve indexicals, reason about the user intent, etc.

Introducing Indexicals

While not a requirement, it would be very user-friendly to allow indexicals in the Dialog grammar. The problem, of course, is that indexicals are often ambiguous. Reasonable assumptions can probably be made, such as the indexical refers to the last statement by the other party. Should a user use an indexical in an ambiguous situation, the AI might make the most reasonable assumption but then make that assumption clear in the response. These are techniques that humans use to make conversation less ambiguous.

Use Case Illustrations: TA1

To further explore options 2 and 3 or requirement 2, use case illustrations may be helpful.

Use Case 1: Extraction of equation from text using only dialog

// extraction from text is happening in the backend

CM: From this text, "An analysis based on conservation of mass and momentum shows that the square of the speed of sound a^2 is equal to the the gas constant R times the temperature T times the ratio of. specific heats gamma. The definition is a^2 = R * T * gamma.", the following were extracted:

(an Equation Eq12345 with expression (a Script with script "R * T * gamma2")).

CM: What is R? What is T? What is gamma2?

(User:)R is the gasConstant of Air.   // AI recognizes this answers one of the previous questions

CM: is T the temperature of the Air?   // the AI is now able to infer this and ask for validation

(User:)Yes.

(User:)script of Eq12345 is "sqrt(R * T * gama)".   // user offers correction to script

(User:)Replace Eq12345 with SpeedOfSound.   // user directs renaming the Equation instance

CM: SpeedOfSound is already an instance of an Equation.   // AI reports name conflict
(User:)Replace Eq12345 with SpeedOfSound2.   // user directs renaming the Equation instance
CM: is gama the gamma of the Air?   // AI seeks confirmation of inference
CM: Equation SpeedOfSound2(decimal R (gasConstant of some Air {"g/mole", "lbm/lbmole"}),
           decimal T (temperature of the Air {Kelvin, Rankine}),
           decimal gamma (gamma of the Air)):
        sqrt(R * T * gamma).   // desired complete specification

Use Case 2: Extraction of equation from text using Web-based interaction with text display

// extraction from text is happening in the backend

CM: Did not extract anything from "An analysis based on conservation of mass and momentum shows that the square of the speed of sound a^2 is equal to the the gas constant R times the temperature T times the ratio of. specific heats gamma. The definition is a^2 = R * T * gamma.".

CM: please review results here.   // interaction follows in the Web interface, after which...

CM: Extracted (an Equation Eq12345 with expression (a Script with script "R * T * gamma2")).

CM: What is R? What is T? What is gamma2?

etc.

Use Case Illustrations: TA2

Use Case 3: Query requiring computation

What is the machNumber of an Aircraft flying at 20000 ft with a speed of 250 mph?

CM: machNumber of the Aircraft is 0.353.

Use Case 4: Modifying existing triples

a. If a triple exists, e.g., "MyAirplane has altitude 23000 ft." and altitude is a functional property, then a second statement "MyAirplane has altitude 25000 ft" should replace the first triple with the second.

b. A second mechanism is a new "update" keyword: "update MyAirplane has altitude 25000 ft", or perhaps "update altitude of MyAirplane to 25000 ft." If the property is multi-valued, e.g, "MyAirplane has color blue, has color yellow." then "update color of MyAirplane to red" would result in an ambiguity error. It would be necessary to say "update color of MyAirplane from yellow to red."

Use Case 5: Making indexical references to previous statements.

a. How does temperature change with altitude?
CM: "three equations compute temperature as a function of altitude, depending on the altitude."
Graph that.

b. What is the temperature when the altitude is 30000 ft?
CM: the temperature of Air at 30000 ft is -47.8 F.
What if it is 38000?
CM: the temperature of Air at 38000 ft is -70 F.
What is the pressure?
CM: the pressure of Air at 38000 ft is 3.0 psi.

References

Leveson, N. G. (2000, January). Intent Specifications: An Approach to Building Human-Centered
    Specifications. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, pp. 15-35. Retrieved from
    SafeWare Engineering: Engineering for a Safer World.