Keeping up in Conversation

CS395 GAI

Spring, 2005

Motto

- If you can keep your head while others around you are losing theirs, you probably designed the scenario.

Overview

- Using deeper processing to simplify conversational NPC design
- Tackling the Bomb Squad scenario
  - Exploiting properties of the scenario design
  - Fragment parsing and ellipsis
- Tackling the Wub Negotiation scenario
  - Exploiting properties of the scenario design
  - Semantic interpretations as canonicalization
  - Values and making the negotiation work

Using the deep(er) Listener processing

- Use (eal::start-interactive-session) instead of (eal::start-eliza-session)
- Look at working memory contents to see what the sentence(s) have been turned into.
  - (ist-Information (ListenerStoryFn test) (beneficiary give1563 cure1763))
  - (ist-Information (ListenerStoryFn test) (listenerStatementAssertion (ListenerStatementFn 2 test) (TheSet (isa exchange1686 FinancialExchange) […]))
- Hint: To speed development,
  - Use (browse-wm) to look over all assertions
  - Call eal::extract-contents-from-text directly to see what the NLU system does on particular sentences.

Scripting a conversational NPC

- You don’t want to do this via ad hoc code
  - Doesn’t scale, unmaintainable.
- Simple models often suffice
  - Recall Zubek’s guest lecture on use of Finite State Machines and Hidden Markov Models
- What’s a quick-but-clean implementation of such a model?

Suggestion for scripting implementation

- Use finite state machine model
  - How to represent state?
  - How to represent state transitions?
  - How to represent what happens at a state?
    - When state is first entered, allow NPC to do something.
    - When state is exited, allow NPC to do something.
- The model doesn’t have to be pure FSM
  - e.g., use local variables to determine when something has gone on too long and transition even if the player isn’t getting it.

A really simple implementation

- Use a variable with symbols as values for the state
  - Each symbol = distinct state of FSM
- Define generic methods that dispatch on the state
  - `(defmethod state-entry-actions ((eql ‘:initial)…))`
  - `(defmethod state-exit-actions ((eql ‘:initial) …))`
  - `(defmethod state-update-action ((eql ‘:initial) …))`
    ;;; Called each time an utterance occurs that leads
    ;;;; back to the same state
  - `(defmethod state-transition ((eql ‘:initial) …))`
    ;;; Returns symbol representing next state. If no
    ;;; transition,returns the symbol for the current state.

Driver loop

- Get next utterance
- Call state-transition with current state, utterance
  - If output = current state, call state-update-action with current state and utterance
  - Otherwise,
    - Call state-exit-actions for current state
    - Call state-entry-actions for next state

How to process the utterance?

- Given this framework, we have constraints on that process
  - What internal updating needs to be done?
    - e.g., lists of things offered, # turns left until the bomb explodes.
  - Does it signal a state transition?
    - e.g., the interpretation process is state-dependent
  - What response, if any, should be generated?

Option 1: Shallow processing

- Use simple pattern matcher on words from Eliza subsystem as trigger for action.
  - Simple to program, but broad coverage is hard to achieve.

```
(defmethod state-transition ((eql ‘:during-negotiation)
  ..)
  (cond ((matches ut ‘(goodbye)) …)
         ;; N.B. matches returns nil if match is :fail.
         …))
```

Option 2: Deeper processing

- Use unification on semantic interpretation of sentences
  - Can use chainer axioms in FIRE, or do it procedurally.
- Recognize relevant conditions in the interpretation
  - What type of event is it?
  - What relationships does it involve?

Partial interpretations are useful

```
(ist-Information (ListenerStoryFn test)
 (listenerStatementAssertion (ListenerStatementFn
    1 test)
  (TheSet
   (wordSenseChoiceSet offer1436
    (TheSet Offering-CommunicationAct GiftGiving
     MakingSomethingAvailable))
   (substitutedProperName "zork"  zork
    zork-thegame)
   (isa sentence1471 DeclarativeUtterance)
   (hasLexicalFeature KDF COMLEX31Lexicon lex i)
   (hasLexicalFeature offer1436 COMLEX31Lexicon
    lex offer)
   (hasLexicalFeature zork1460 COMLEX31Lexicon lex
    zork)
   (hasLexicalFeature punc-period1468
    COMLEX31Lexicon lex
    punc-period)
   (isa si3326027616 SemanticInterpretation)
   (memberOfSemanticInterpretation si3326027616
    si3326027616)
   (myCreator si3326027616 Kenneth Forbus)
   (myCreationPlace si3326027616 shalmaneser)
   (myCreationTime si3326027616 3326027616)
   (textOfSentence sentence1471 "I offer Zork." )
   (sentenceInSemanticInterpretation sentence1471
    si3326027616)
   (textForSemanticInterpretation si3326027616
    "I offer Zork." ))))
```

- The wordSenseChoiceSet gets the gist of the verb
- The proper name substitution tells you that a recognized entity was mentioned.
- Context lets you put a reasonable interpretation on it
- Hint: Keep sentences simple
  - “We will give you Zork in exchange for the cure.”

Tackling the Bomb Squad

- What scenario constraints can we exploit?
  - IM interface → telegraphic communication
    - Long rambling sentences less likely.
    - Annoyance: Contractions are not handled correctly by Listener
  - The Player is the Bomb Squad expert
    - The player is given the specifications for the bomb
      - Which means we know them, too, and can use this in scripting the NPC
    - The NPC controls forays into other realms
      - We can have reasonable expectations about the general kinds of soothing remarks the player will make.
  - The dynamics of the situation call for rapid, short conversations

Ellipsis

- Recall SOPHIE:
  - “What was the voltage at N1?”
    “N2?”
- Possible solutions:
  - Implement a general ellipsis mechanism
  - Exploit script context

Ellipsis: A general solution

- Keep the parse of the previous sentence around.
- Given a single fragment as an utterance, determine what it might correspond to in the previous sentence.
  - Use parse tree to find plausible candidates.
  - Run semantic interpreter to filter
- If successful, treat the utterance as if it was that sentence
  - If multiple possibilities, ask player which they meant.

Ellipsis: A quick hack

- Model script as a finite state machine
  - Recall Zubek's guest lecture on dialogue via FSMs and HMMs
- For each state, possible fragment responses also treated as transitions

Processing Fragments

- The parser will handle phrases and fragments of sentences, too.
- Use context to provide expectations that enable you to interpret fragments.

Example of using fragments

```
(ist-Information (ListenerStoryFn test)
 (listenerStatementAssertion
   (ListenerStatementFn 3 test)
  (TheSet (isa wire1951 Wire)
   (hasMentalAttributes wire1951
    (MediumToVeryHighAmountFn Sadness))
   (mainColorOfObject wire1951
   BlueColor)
   (feelsEmotion wire1951
    (MediumToVeryHighAmountFn Sadness))
   (feelsEmotionTypeAtLevel wire1951
   Sadness
    MediumToVeryHigh)
   (hasAttributes wire1951 BlueColor)
   (isa BlueColor SensoryAttribute)
   (isa BlueColor ChromaticColor) (isa
   BlueColor Color)
   (isa BlueColor PhysicalAttribute)
   (isa BlueColor
   CompositeScalarInterval)
   (isa BlueColor IKBConstant)
   (wordSenseChoiceSet wire1951 (TheSet
   Wire Wire))
   (wordSenseChoiceSet blue1923
    (TheSet BlueColor
   (MediumToVeryHighAmountFn
   Sadness)))
```

```
[…]
  (hasLexicalFeature punc-period1969
  COMLEX31Lexicon lex
   punc-period)
  (isa si3326028178
  SemanticInterpretation)
  (memberOfSemanticInterpretation
  si3326028178
   si3326028178)
  (myCreator si3326028178 Kenneth
  Forbus)
  (myCreationPlace si3326028178
  shalmaneser)
  (myCreationTime si3326028178
  3326028178)
  (textOfSentence sentence1971 "The
  blue wire." )
  (sentenceInSemanticInterpretation
  sentence1971
   si3326028178)
  (textForSemanticInterpretation
  si3326028178
   "The blue wire." ))))
```

Another Example

```
(ist-Information (ListenerStoryFn test)
 (listenerStatementAssertion
    (ListenerStatementFn 4 test)
  (TheSet
   (missingSemTrans blue1919 noun Blue-
   TheWord)
   (missingSemTrans one2011 pronoun One-
   TheWord)
   (hasLexicalFeature the1972
   COMLEX31Lexicon lex the)
   (hasLexicalFeature blue1983
   COMLEX31Lexicon lex blue)
   (hasLexicalFeature one2011
   COMLEX31Lexicon lex one)
   (missingSemTrans punc-period2023 punc
    Punc-Period-TheWord)
   (hasLexicalFeature punc-period2023
   COMLEX31Lexicon lex
    punc-period)
   (isa si3326028592
   SemanticInterpretation)
   (memberOfSemanticInterpretation
   si3326028592
    si3326028592)
   (myCreator si3326028592 Kenneth
   Forbus)
   (myCreationPlace si3326028592
   shalmaneser)
   (myCreationTime si3326028592
   3326028593)
   (textOfSentence sentence2025 "The blue
   one." )
   (sentenceInSemanticInterpretation
   sentence2025
    si3326028592)
   (textForSemanticInterpretation
   si3326028592
    "The blue one." ))))
```

- You can recognize cues in the semantic interpretation more easily than in the word string
  - "Do I cut the blue wire or the green wire?" versus "Cut the green or the blue?"
  - "Clip the green or the blue?" versus "Clip the green? Or the blue?"
- Important point: When the NLU system fails to find a full interpretation, the partial information it provides will often be enough.

Tackling the Wub Negotiation

- Exploiting properties of the scenario design
  - IM channel + Universal translator
    - Misunderstandings routine
    - Finding workable language subset important as part of gameplay
  - Alien
    - It is likely to violate our expectations in many ways.
    - Scamming it is part of the fun.
      - Although, what will it do with those Barney videos?

Semantic interpretations as canonicalization

- Recall previous Zork example
- Same principles as in Bomb Squad
- Some of the details will be different:
  - Recognizing cultural artifacts will be important.
  - Characterizing different kinds of utterances, in addition to content
    - Adding an item to the offer
    - Removing an item from the offer
    - Questioning as to possible interests

Values

- The Wub needs to set a price.
  - – Shopping list of specific items?
    - Player would have to figure out these items.
  - – Internal value scale?
  - – Need a model of how much it values particular artifacts and kinds of artifacts, plus its technologies on offer.

Making the negotiation work

- Orienting the player
  - Suppose no interesting offer in several turns?
    - Might want to give them a hint or even an example of the kind of thing that would be of interest.
- How hard a bargain to drive?
  - What is the artistic goal?
    - Extract particular things from the player?
    - Extract the most value from the player?
    - Amuse the player (via value skew)?
    - Keep them engaged for a particular length of time?
    - Reveal story background as a side-effect of the conversation?