Dialogue systems

Volha Petukhova
Saarland University
Introduction

Multimodal natural-language based dialogue as human-machine interface
Introduction: dialogue systems
· https://www.youtube.com/watch?v=5FFRoYhTJQQ
· https://www.youtube.com/watch?v=THPSzpm0Zug
· https://www.youtube.com/watch?v=zI FMq5IWVjI
· https://www.youtube.com/watch?v=rg0fmaCSDPg
Tasks of Dialogue Management

● Dialogue flow control

● Dialogue modeling
  ➔ Dialogue context
  ➔ Dialogue acts

● Dialogue act decision making

● Dialogue phenomena:
  • Error handling
  • Initiative and cooperation
  • Adaptivity
  • – ...
Dialogue Management: approaches

- **Script-based (state machines)**
  
  Sequence of pre-defined steps (dialogue script)

- **Frame-based (also: form-filling)**
  
  Set of slots to be filled (task template) and corresponding prompts

- **Plan-based**
  
  Collaborative problem solving

- **Generic paradigm: Information-State Update**
  
  Declarative rules for updating dialogue context
Script-Based DM

- Script describes all possible dialogues
- Typically finite state machine

- Set of states and transitions
  - State determines system utterance
  - User utterance determines transition to next state (deterministic)
- No recursion! (= no nested sub-dialogues)

- Fixed dialogue script
- OK for system-driven interaction
Finite State Machine

- <States, Init-State, Alphabet, Transition-function>
- Variants: machines having
  - actions associated with states (Moore machine)
  - actions associated with transitions (Mealy machine)
  - multiple start states
  - transitions conditioned on no input symbol (a null)
  - more than one transition for a given symbol and state (nondeterministic finite state machine)
  - states designated as accepting states (recognizer)
  - etc.

See, e.g., NIST http://www.nist.gov/dads/HTML/finiteStateMachine.html
FSM-Based Models

\[ U: \text{Elevator?} \]
\[ S: \text{Hello. Which floor would you like to go to?} \]
\[ U: \text{Third floor.} \]
\[ S: \text{OK, I am taking you to the third floor.} \]

States: ...
Init-State: ...
Alphabet: ...
Transition function: ...
FSM-Based Models

U: Elevator?
S: Hello. Where would you like to go to?
U: Prof. Barry.
S: Prof. Barry is on the fourth floor.
I am taking you to the fourth floor.

Extension: variable for floor number
FSM-Based Models

States: ...
Init-State: ...
Alphabet: ...
Transition function: ...

[McTear 2002]
FSM-Based DM: Sum Up

• Advantages
  ▪ Fixed prompts can be pre-recorded
  ▪ Speech recognition and input interpretation can be tuned for each state

• Disadvantages
  ▪ Rigid dialogue flow
  ▪ Inhibiting user initiative
  ▪ Only suitable for simple tasks
  ▪ In principle can make more flexible, but it quickly gets very complex; modular solutions are possible
CSLU toolkit

Frame-Based DM (Form Filling)

- Frame (form): what info should be supplied by user

  \[
  \begin{array}{l}
  \text{departure\_city} \, ? \\
  \text{departure\_date} \, ? \\
  \text{destination\_city} \, ? \\
  \text{return\_date} \, ? \\
  \ldots \\
  \end{array}
  \]

- Dialogue states: which slots are filled

- General routines for what system should do next (given which slots are filled)
Frame-Based Models

S: Where do you want to go?
U: Paris

S: Where will you travel from?
U: From Berlin.

S: When will you travel?
U: August 1st.
Frame-Based Models

S: What can I do for you?
U: I want to fly to Paris

S: Where will you fly from?
U: From Berlin on August 1st.

“Overanswering”
Frame-Based Models

• Strategies for deciding what to do next
  ▪ Next unfilled slot
  ▪ Slot-combination weighting
  ▪ Ontology-based coherence

• Options for database lookup
  ▪ Delayed (typically; after certain slots filled)
  ▪ Immediate (can be “expensive” = take time, but enables more helpful system behavior)
VoiceXML

• Digital document standard for specifying interactive media and voice dialogue
• Used for developing audio and voice response applications
• VoiceXML documents are interpreted by a voice browser
• VoiceXML application collects and processes info, and plays back info
VoiceXML

- Main elements of a VoiceXML document
  - Form: basic unit of functionality
  - Field: prompts for and accepts user input
  - Prompt: sequence of audio elements or TTS messages
  - Audio: audio file or TTS message to play
  - Filled: processes input, can pass control to other forms

- Form Interpretation Algorithm
  - Defines how fields in a form are filled in, and how the fill ordering can be modified

- Global event handlers (e.g., for error handling, help)
  - Define behavior when predefined global conditions occur

- Control transfer conditions and subroutine constructs (= special-purpose programming language)

- new, more expressive standard: State Chart XML
VoiceXML

See VoiceXML tutorials

http://www.vocomosoft.com/voicexml_tutorial.htm

Some tools (not exactly VoiceXML, but close)

www.wit.ai

https://www.luis.ai/
Frame-Based DM: Sum Up

- **Advantages**
  - More flexible dialogue
  - Enables some user initiative

- **Disadvantages**
  - Speech recognition more difficult, because user input less restricted
  - Not every task can be modeled by a frame
Plan-Based DM

• Communication is a joint activity: Agents communicate to establish common ground, agents collaborate to accomplish a task

• Collaborative problem solving by (rational) agents
  ▪ Neither agent can accomplish the task alone
  ▪ Need **joint goals** and **mutual understanding**
  ▪ Agents collaborate to establish and achieve their goals

• Agents have knowledge about solving tasks
  ▪ deciding on goals (objectives): adopt, select, defer, abandon, release
  ▪ Forming plans to achieve goals (recipes)

• Automated planning: STRIPS; planning operators: actions, preconditions, post-conditions
  ▪ Executing plans (acting)
  ▪ Revising decisions (re-planning, abandoning goals, etc.)

• Agents reason about beliefs and actions
• Intention recognition
Plan recognition

Given: plan for getting a BA

U: I’ll take German 101 fall semester.
User: Send ambulance one to Parma right away.
  (initiate (c-adopt (action (send amb1 Parma))))
  (initiate (c-select (action (send amb1 Parma))))
System: OK. [sends ambulance]
  (complete (c-adopt (action (send amb1 Parma))))
  (complete (c-select (action (send amb1 Parma))))
System: Where should we take the victim once we pick them up?
  (initiate (c-adopt (resource (hospital ?x))))
User: Rochester General Hospital.
  (continue (c-adopt (resource (hospital RocGen))))
System: OK.
  (complete (c-adopt (resource (hospital RocGen))))

[Blaylock et al. 2003]
Plan-based DM

• Advantages
  ▪ Flexibility and adaptivity
  ▪ Any task can be modeled
  ▪ … the ultimate solution

• Disadvantages
  ▪ Specifying planning operators is as hard as writing dialogue scripts
  ▪ Plan recognition is a hard problem
  ▪ Lots of reasoning needed
Dialogue Modeling as Information State Update

IS $\rightarrow$ IS'

Diagram with boxes labeled IS and IS' connected by an arrow.
Information State

• Representation of the current state of dialogue
• Used by system to
  - Interpret user’s contribution
  - Decide which actions to take
  - Decide what to say
  - Store information (dialogue context representation)
• Utterances update information state
• Approaches to DM differ in how IS is represented, what role it plays, what it contains
ISU Dialogue Modeling

Components:

– a description of the informational components of the IS (aspects of common context, participants, common ground, linguistic and intensional structure, commitments, beliefs, intentions, user model...)

– their formal representation (e.g. lists, sets, typed feature structures, DRSs, propositions, modal operators, etc.)

– set of dialogue acts (DAs) triggering the update of the IS

– set of update rules governing the IS updates given various conditions of current IS and performed DAs (e.g. set of selection rules that license choosing a particular DM to perform given IS)

– a control strategy to decide which update rule(s) to select at a given point in the dialogue (e.g. „pick first that applies”, game theory, statistical methods)
IS Update Rules

Describe possible transitions from one information state to the next

If <conditions-on-IS-values>
then <changes-to-IS-values>

Conditions: when a rule is applicable
Effects: how the IS changes

IS1  A: What time is it?  IS2  B: 10 o’clock.  IS3
State Machine Model as ISU

- IS: current-state; input
- Update rules:
  
  If [state] & [input]
  
  then [output]; [next-state]
Frame-Based Model as ISU

- **IS**: task-frame; user’s move; system move
- Update rules: e.g.,

  If [user move = slot X value V] then [fill X with V]

  If <conditions-on-frame-values>

  then <ask-slot-value Y>

Decision about next system move is also a rule
ISU-Based Dialogue Modeling

• Task- vs. Dialogue-Structure
  • Task --> dialogue
  • But, dialogue does not have to follow task (execution) structure

• “Dialogue planning”: creating an agenda
  • Task model fills agenda with task-related goals
  • Dialogue manager can add more goals, e.g., for grounding

• Some approaches:
  • QUD-based: Godis (TRINDI, SIRIDUS)
  • Obligation-based: Edis (TRINDI)
  • Agent-based: collaborative problem solving: TALK
Chatbots

A chatbot is a conversational agent that interacts with users using natural language.

- First chatbot - ELIZA (Weizenbaum 1966), which emulated a psychotherapist: http://nlp-addiction.com/eliza/
- ALICE is a chatbot: **ALICE System**
  http://www.alicebot.org/about.html
- **ALICE**: the Artificial Linguistic Internet Computer Entity; a software robot that you can chat with using natural language.
- ALICE language knowledge is stored in **AIML** files.
- **AIML**: The Artificial Intelligence Mark up Language.
Topics: each Topic file contains a list of categories
Categories: contain
  - Pattern: to match with user input
  - Template: represents ALICE output

<aiml version="1.0">
<topic name="the topic">
  <category>
    <pattern>PATTERN</pattern>
    <template>Template</template>
  </category>
..</topic>
</aiml>
• CATEGORIES (Basic unit of knowledge)
  » <category>
    <pattern>HELLO</pattern>
    <template>Hi there!</template>
  </category>

• Consists of: Input Question, Output Answer, [Context]
  • Pattern = Initial question (a.k.a. “Stimulus”)
  • Template = Answer (a.k.a. “Response”)
  • Context = Optional, “that” or “topic”

• Consists only of words, spaces and wildcards _ and *
  • Words have letters and/or numerals, space separated
  • Case insensitive
If you're new to Pandorabots and AIML, you should first try customizing your pandorabot by changing some of its properties or by providing your own custom responses with the training interface.

For more advanced botmasters, this page allows you to download, modify and upload the AIML files for your pandorabot directly.

The tables below show all the AIML files for this pandorabot. To view or edit a file, click on its name.

<table>
<thead>
<tr>
<th>Filename</th>
<th>Size</th>
<th>Last Modified</th>
<th>Browse</th>
<th>Download</th>
<th>Library</th>
<th>Active</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALI.aiml</td>
<td>32,931</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>ALICE.aiml</td>
<td>36,385</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Astrology.aiml</td>
<td>1,227</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Atomic.aiml</td>
<td>427,931</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Biography.aiml</td>
<td>84,535</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Bot.aiml</td>
<td>415,844</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Botmaster.aiml</td>
<td>5,350</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Client.aiml</td>
<td>138,096</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Computers.aiml</td>
<td>21,925</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Drugs.aiml</td>
<td>11,905</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Emotion.aiml</td>
<td>15,784</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Food.aiml</td>
<td>4,901</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Geography.aiml</td>
<td>167,419</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>History.aiml</td>
<td>1,309</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Humor.aiml</td>
<td>1,215</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Inquiry.aiml</td>
<td>3,298</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Interjection.aiml</td>
<td>2,613</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>IQ.aiml</td>
<td>1,369</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Knowledge.aiml</td>
<td>266,551</td>
<td>09/15/2003 06:05:17 PM</td>
<td>Browse</td>
<td>Download</td>
<td>aaa-09-2003</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
Let’s talk to Alice
Virtual Human toolkit: NPCEditor