Computational Dialogue Models

Volha (Olga) Petukhova
Spoken Language Systems Group
Saarland University
Outline

Introduction
Dialogue Systems | Examples

Dialogue System Architecture
Components | Tasks

Dialogue Management
Script-based | Frame-based | Plan-based | Information State Update | Agent-based | ChatBots

Evaluation
User-based evaluation | Metrics

Development toolkits
CSLU | LUIS | Virtual Human | OpenDial
Introduction

Multimodal natural-language based dialogue as human-machine interface

Computational Pragmatics, Winter 2019/2020

Petukhova
Examples

- https://www.youtube.com/watch?v=zlFMq5IwVjI
- https://www.youtube.com/watch?v=t7Krn-DH3tw
- https://www.youtube.com/watch?v=YZizCoOctPo
Dialogue Systems (general architecture)

Input Modules
- ASR
- Visual
- Touch/haptic

Interpret

Fusion

Discourse Model

Dialogue Manager

Output Modules
- TTS
- Visual

Generate

Fission
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 Modelling: 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

**Information-State Update**
Declarative rules for updating dialogue context

**Statistical (PO)MDP-based models**
Probability distribution of the events or user states observed/learned from the observed past

**End-to-End models**
sequence2sequence models learned from large amount of data
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
<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: Elevator?
S: Hello. Which floor would you like to go to?
U: Third floor.
S: OK, I am taking you to the third floor.

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

init
Welcome
Ask_floor
Floor_n
Floor_1
Not_und

$\epsilon$
floor n
floor 1
unknown
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 Models

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

[McTear 2002]
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
Frame-Based DM (form filling)

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

```
departure_city?
departure_date?
destination_city?
return_date?
```

…

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.

departure_city
departure_date
destination_city
return_date

departure_city  ?
departure_date  ?
destination_city  Paris
return_date  ?

...
Frame-Based Models

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

<table>
<thead>
<tr>
<th>departure_city</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>departure_date</td>
<td>?</td>
</tr>
<tr>
<td>destination_city</td>
<td>Paris</td>
</tr>
<tr>
<td>return_date</td>
<td>?</td>
</tr>
</tbody>
</table>

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

“Overanswering”

<table>
<thead>
<tr>
<th>departure_city</th>
<th>Berlin</th>
</tr>
</thead>
<tbody>
<tr>
<td>departure_date</td>
<td>1/8/05</td>
</tr>
<tr>
<td>destination_city</td>
<td>Paris</td>
</tr>
<tr>
<td>return_date</td>
<td>?</td>
</tr>
</tbody>
</table>

...
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)
Frame-Based Models: 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
Communication is a **joint activity**: participants communicate to establish common ground, participants 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**

- Decide on goals (objectives): adopt, select, defer, abandon, release
- Form plans to achieve goals (recipes)
Plan-based Models

Automated planning: STRIPS; planning operators: actions, reconditions, 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.

Diagram:

```
AcademicPlan
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
GetBA(PolySci)  | Math/LangReq  |
  |               |
GetGermanCredits
  |               |
German101(fall03)
```
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))))
Plan-Based DM: sum up

- 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
Information State Update

\[ IS \xrightarrow{\text{Interpretation of behaviour}} IS' \]
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
Components:

- a description of the information state 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>

IS1 \[\rightarrow\] A: What time is it? \[\rightarrow\] IS2 \[\rightarrow\] B: 10 o’clock. \[\rightarrow\] IS3
S: It’s raining outside

**preconditions:**

\[ \text{Bel}(S; p) \]
\[ \text{Want}(S; \text{Bel}(A; p)) \]

**expected understanding:**

\[ \text{Bel}(S, \text{MBel}({S, A}, \text{WBel}(S, \text{Bel}(A; \text{Bel}(S, p)))))) \]
\[ \text{Bel}(S, \text{MBel}({S, A}, \text{WBel}(S, \text{Bel}(A; \text{Want}(S; \text{Bel}(A; p))))))) \]

**expected adoption:**

\[ \text{Bel}(S, \text{MBel}({S, A}, \text{WBel}(S, \text{Bel}(A, p)))) \]
A: no, it isn’t

understanding:

\[
\text{Bel}(A, \text{MBel} \{S,A\}, \text{Bel}(S,p)))
\]

\[
\text{Bel}(A, \text{MBel} \{S,A\}, \text{Want}(S, \text{Bel}(A, p)))
\]

adoption:

\[
\text{Bel}(A, \text{MBel} \{S,A\}, \text{Bel}(A,p)))
\]

preconditions:

\[
\text{Bel}(A, \neg p)
\]

\[
\text{Want}(A, \text{Bel}(S, \text{Bel}(A, \neg p)))
\]

expected understanding:

...

expected adoption:

...
A: yes, it is

understanding:

\[ \text{Bel}(A, \text{MBel} \{S,A\}, \text{Bel}(S,p))) \]
\[ \text{Bel}(A, \text{MBel} \{S,A\}, \text{Want}(S, \text{Bel}(A, p)))) \]

adoption:

\[ \text{Bel}(A, \text{MBel} \{S,A\}, \text{Bel}(A,p))) \]

preconditions:

\[ \text{Bel}(A, p) \]
\[ \text{Want}(A, \text{Bel}(S, \text{Bel}(A, p))) \]

expected understanding:

...

expected adoption:

...
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 Modelling

- 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)
Dialogue Contribution Processing

Dialogue Act Agents
- Task-Agent
- AutoFB-Agent
- AlloFB-Agent
- TurnM-Agent
- TimeM-Agent
- etc.

Context model
- Cognitive context
- Semantic context
- Social context
- Perceptual context
- Linguistic context

Dialogue act recognition/generation
- DA1
- DA2
- DA3

Features selection/clustering/classification
- CL1
- CL2
- CL3

Features
- wording
- syntax
- prosody
- nonverbal
- previous utterances features
- utterance

Update operators

Candidate DAs

Context Manager
ChatBots

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


- 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)

```xml
<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”
Yak >> AIML files

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>Alice.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>167,418</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>1,389</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>I.U.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>
Virtual Human Toolkit: NPCEditor

![NPC Editor Screenshot]
LUIS (Microsoft)

https://www.luis.ai/home
http://www.opendial-toolkit.net/
CSLU toolkit

https://www.youtube.com/watch?v=ZrAlj7GQqjQ