An Introduction to Dialogue Systems

Oliver Lemon
olemon@inf.ed.ac.uk

This presentation will probably involve audience discussion, which will create action items. Use PowerPoint to keep track of these action items during your presentation:

• In Slide Show, click on the right mouse button
• Select “Meeting Minder”
• Select the “Action Items” tab
• Type in action items as they come up
• Click OK to dismiss this box

This will automatically create an Action Item slide at the end of your presentation with your points entered.
Outline

- Overview of dialogue systems research
- Focus on “Practical”/Task-based dialogue systems:
  - Dialogue management
  - Task modelling
- A case study: WITAS demonstration system and its components
  - OAA, GEMINI, Nuance, Festival
- Some research issues
  - Context-sensitive speech recognition
  - User studies and “Targeted Help” error handling
What is a DS?

- An interface that supports natural language input, responds using natural language, and emulates some human conversational skills:
  - Recognising dialogue structures
  - Contextual interpretation
  - Turn taking
  - Managing conversational obligations
  - Response planning
  - ......
  - http://www.sigdial.org/
Sci-fi dialogue systems

- “HAL 2000”: full dialogue competence, lip reading, singing…..
- Star Trek ship’s computer: must be addressed by name, limited functionality
- Start Trek’s “DATA”: no personality
- Red Dwarf’s “Holly”: different personalities
- all have continuous speaker-independent speech recognition and emulate human conversational behaviour
- all maintain a representation of dialogue context, execute conversation plans and skills
State of the art 2003

- Form-based and FSM “speech-driven” applications in commercial use:
  - automated call routing
  - stock trading
  - bill paying (e.g. Telewest Broadband)

- More flexible dialogue systems are research prototypes:
  - TRAINS – finding effective routes on a map
  - TRIPS – deployment and scheduling tasks
  - WITAS – instructing and monitoring tasks of a mobile robot
  - Godot – dialogues with a robot
  - Personal Satellite Assistant (NASA)
  - BEETLE – tutorial dialogues
2003: Types of application

- Information seeking and form filling:
  - Flight booking, bill pay
  - Booking cinema tickets...

- Command and control:
  - Robots, devices, games, ...

- Tutorial systems:
  - Teaching electronics, physics, maths...

- “Believable agents”
  - Chatbots, salespeople, newscasters, characters...

- Spoken and Multimodal systems:
  - Speech, text, and/or gestures?
Main issues

- Dialogue management
  - Modelling dialogue context, dialogue “moves” (e.g. clarification questions), updating “information states”
  - Determining meaning in context, e.g. anaphora
  - Intention recognition, negotiation, collaboration
  - Dialogue as collaborative problem solving
  - Initiative – taking and releasing the floor

- NLG: what to say next, when to say it, and how to say it
  - Content planning, sentence planning, lexical choice
  - Turn taking
  - Speech synthesis and intonation
Dialogue modelling approaches

- **Finite state dialogue models**
  - Dialogues described as “graphs” through possible conversations to an end state (e.g. “voice buttons”)
  - Simple, sufficient for very controlled/scripted types of interaction. Not true conversation.

- **“Information State Update” (ISU) approach**
  - More flexible, more complex, more realistic

- **Task-based approaches**
  - Planning and plan recognition
  - Domain models
  - Constraint reasoning
FSM based Dialogue Management

- They can:
  - Run in real time
  - Produce help sensitive to task context
  - Re-prompts sensitive to task context

- Hard to:
  - Make response sensitive to unexpected input
  - Make response sensitive to linguistic context
  - Provide personalised/customised advice or feedback
Designing FSM based DM

DM design is labour intensive, domain specific, and error prone:

- size is practically, not theoretically, limited
  - typical banking application has 1500 states
- need to specify what to do in a wide range of cases for each state
  - e.g., help, cancel, timeout
- design all dialogue flow control manually
  - alternative routes must be drawn in full
Some tools/systems

- **Finite state:**
  - CSLU toolkit

- **Information state update (ISU):**
  - DIPPER: [http://www.ltg.ed.ac.uk/dipper/](http://www.ltg.ed.ac.uk/dipper/)
  - BEETLE

- **Task based/“Practical”:**
  - TRIPS/TRAINS
  - NASA’s Personal Satellite Assistant
  - WITAS (based on ISU approach)
Task-based DS

- Most research is carried out in the genre of “information” dialogues, e.g. booking a flight.
- Form-based systems manage sequential tasks (e.g. get destination, get date, …)
  - but what if we are managing concurrent tasks? with no end state?
    - with duration, states, and constraints?
    - in a dynamic environment?
    - where dialogues cannot be scripted?
- Most dialogue managers are domain-specific
  - but what if we want to re-use/re-configure the dialogue system for a different processes?
“Practical Dialogue”

(James Allen, et al, 2001)

- “Practical” – in order to achieve tasks.
- The Practical Dialogue Hypothesis: “The conversational competence required for practical dialogues, while still complex, is significantly simpler to achieve than general human conversational competence”
- The Domain-Independence Hypothesis: “Within the genre of practical dialogue, the bulk of the complexity in the language interpretation and dialogue management is independent of the task being performed”
Software engineering issue:

- Task-oriented devices/agents have domain-specific abilities which we want to coordinate using a generic dialogue competence.
- Can we capture general features across a range of devices which allow a single dialogue front-end to be easily adapted to work with all of them?
- Activity modelling e.g. WITAS “Robocopter”
Build a (multimodal) dialogue system to support human interaction with autonomous devices:
- E.g. robot helicopter (WITAS project’s UAV)
- Human and robot negotiate multiple “activities” with constraints

- Multimodality (open-mic, mouse clicks, text)
- Interleave planning and execution dialogues for multiple activities
- Speed issues (recognition, parsing, interpretation in context, generation)
- Robustness of interactions (ambiguity resolution, revision/repair, cancel …)
Research aims for WITAS/Stanford dialogue system:

- More “conversational” dialogue systems:
  - Not voice-buttons, FSMs, or forms
    - Don’t constrain the user’s input choice (unscriptable, open-ended)
    - Co-ordinate joint activities/tasks
    - More natural generation and turn-taking

- Portability and modularity:
  - Current systems use 100s of domain-specific DM rules (e.g. MIT Pegasus etc.)
  - We should re-use dialogue move objects and context update methods
  - Build domain-specific “Task models” for individual devices and services
Multi-tasking and collaboration

- U: “Our job is to search for a red car”
- R: “I will look for a red car”
- U: “Fly to the tower”
- R: “I will fly there and look for a red car”
- R: “I am taking off”
- U: “What are you doing now?”
- R: “Now flying to the tower”
- R: “I see a red car on Circle Road” [GUI]
- R: “Is this the right car?”
- U: “Maybe. Zoom in”
- ............
Constraints and negotiation

- U: “Always fly at high speed”
- R: “Okay, I will always fly quickly”
- U: “Fly slowly to the school”
- R: “Wait a minute. I am supposed to always fly quickly”
- R: “Shall I fly there slowly anyway?”
- U: “Yes”
- R: “OK. Now flying to the school at low speed”
- U: “Fight the fire at the tower”
- R: “Wait a minute. Fighting the fire at the fire conflicts with flying to the school.”
- R: “Shall I fight the fire now or later?”
- …………. → DEMO
More information.....

- http://homepages.inf.ed.ac.uk/olemon/
- http://www-csli.stanford.edu/semlab/witas/
Main Ideas

- “Multi-threaded” dialogue context
  - Tree rather than stack
  - Less constraining of the user
  - Context-sensitive Language Models for SR
- Represent tasks, their structure and states (current, failed, complete, …) – Activity Tree
- Represent local and global constraints.
- Domain-specific Task Models (‘recipes’)
  - Search for X, fly to Y, fight-fire at Z, deliver A to B
  - Teach procedure X, teach rule Y, review actions
  - Play CD, mp3
  - Plan a meeting, read email …….
System Architecture

Dialogue Manager
(context modeling, anaphora resolution, NL generation)

GUI
Speech Recognition
Parser
TTS
Databases

Process Models/Recipes
Joint-Activity Interface
Task State Model (Activity Tree)
Constraint Management System
Device
Walkthrough

- User: “Fly to the tower” → speech recognizer → Parser
- Parser: command([go], param_list([pp_loc (to, arg([np(det([def],the),[n(tower,sg)])))]))
- DMT: attach command node to root node, make it the most active node, add <the tower (speech)> to Salience List
- Resolve NP “the tower” (presuppositions: exists? unique?)
- Search the Activity Model for the go task
- Add go task to Activity Tree, with args
- Begin go task, add report to System Agenda → NLG
- Monitor task progress, listen for user input
- Process System Agenda: take turn, say report (generate NP, use anaphoric expression) ……..
Collaborative Task model: “search”

process Search {ResourcesUsed {camera}
  RequiredProperties {THIS.command=="find");}
  PreConditions {(Status flight inair), // KIF
                    (Status camera ok)} // for JTP
  SkipConditions {(Status locked-on THIS.np)}
  PostConditions {(Status locked-on THIS.np)}
  Children SEQ {TaskProperties {
                command = "locate"; np = THIS.np;
                TaskProperties {
                command = "ask_done";       }}}
}
Other domains:

- In-car, navigation etc (TALK project, www.talk-project.org)
- Automated home (TALK project, d’Homme)
- In-car mp3 player (Bosch RTC)
- Personal organizer (CALO, scheduling meetings)
- Dialogues between groups and teams
- Game agents, interactive learning, …..?
Conclusions

- Limitations of finite state approaches
- Flexibility of “Information State Update” approach (TRINDI, DIPPER)
- Genre of “Practical dialogues”
- Importance of domain-general approach to practical dialogue management
- Convergence on architectures with “activity models” and task representations (e.g. COLLAGEN, WITAS)
- Overview of a dialogue system, its components, representations, and algorithms
- Ability to generalise context-sensitive speech recognition for the ISU approach
Dialogue Systems at Edinburgh

- DIPPER dialogue system toolkit
  - Based on TRINIDIKIT
  - [http://www.ltg.ed.ac.uk/dipper/](http://www.ltg.ed.ac.uk/dipper/)
  - Godot the robot: [http://www.ltg.ed.ac.uk/godot/](http://www.ltg.ed.ac.uk/godot/)
  - D’Homme: automated house

- BEETLE – tutorial dialogue system
  - [http://www.cogsci.ed.ac.uk/~jmoore/tutoring/](http://www.cogsci.ed.ac.uk/~jmoore/tutoring/)

- TALK - Machine learning and dialogue
  - [http://www.talk-project.org](http://www.talk-project.org)
  - olemon@inf.ed.ac.uk
Publications

- Detailed paper in TAL special issue (Lemon, Gruenstein, Peters 2002)
- SIGdial 2002 (Lemon et al.)
- EACL 2003 (Hockey et al.)
- EACL 2003 dialogue system workshop (Lemon and Cavedon)
- AAAI 2003 NLG in dialogue workshop (Lemon et al.)
- SIGdial 2003 (Lemon, Cavedon, Kelly)