Discourse and Dialogue Models

James Pustejovsky

August 27, 2020

Why build dialogue systems?

- Theoretical purpose: test theories
  - e.g. what kind of information does an agent need to keep track of in order to be able to participate in a dialogue?
  - However, complex system with many components – how to evaluate
- Practical purpose: human-computer interaction

Why spoken interaction?

- Spoken interaction is the natural way for humans to interact
  - computers should adapt to humans rather than the other way around
  - important to enable systems to interact in a natural way
- Language can be used to convey any message, at any time
  - On a screen, you can only push the buttons shown
  - Less effort for user, who can just say what's on her mind...
  - ...but system then needs to be able to deal with most of the ways that the dialogue may unfold
- Users want hands-free and/or eyes-free use
  - Especially in in-vehicle situations

History of dialogue systems

- ELIZA (Weizenbaum 1966)
  - text dialogue
  - simulated psychoanalyst
- SHRDLU (Winograd 1972)
  - written dialogue
  - control simulated robot in a blocks world
- TRAINS (Allen et al 1991)
  - spoken dialogue
  - joint planning task
- CSLU Toolkit (McTear 1993)
  - platform for implementing dialogue system applications
  - simple dialogue manager
- Philips train timetable system (Aust et al 1994)
  - speech over phone
  - first deployed system
- Linguatronics (1996)
  - in-car spoken dialogue
  - dialing etc
- VoiceXML (W3C 2000)
  - general platform
  - form-filling dialogue
- Siri (Apple 2009)
  - smartphone-based
  - multimodal
- API.AI, Amazon Alexa (2015)
  - proprietary platforms open for third party development

Two types of methods in Computational Linguistics

- Rule-based
- Statistical/Machine Learning

Rule-based methods

Example: Interpret English commands in infotainment system

- create a lexicon for English
- write grammar rules for English in the infotainment domain
- write rules relating English sentences to a semantic representation (intents and entities)

Statistical/Machine Learning methods

Example: Interpret English commands in infotainment system

- collect lots of examples of English sentences from the infotainment domain
- annotate sentences with their meanings (intents and entities)
- use machine learning techniques to produce statistical models correlating English sentences with intents and entities

Comparing rule-based and statistical methods

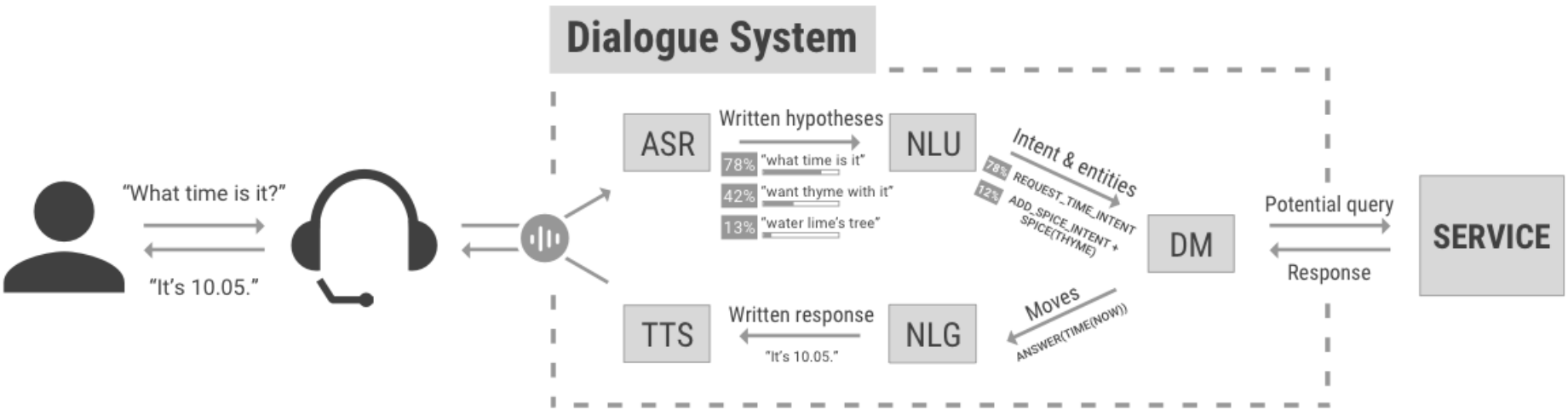
- Rule-based methods get more exact and correct results, but it can take a lot of work to get them to cover enough data
- Statistical methods cover a lot more data, but they sometimes get things very wrong, in ways that we do not understand

Hybrid systems

- Hybrid systems attempt to combine both rule-based and statistical methods
- . . . but there are many open research questions concerning the best way to combine the two approaches

Outline

Dialogue systems architecture



Automatic Speech Recognition (ASR)

- a.k.a. Speech To Text
- Acoustic model: sound to phonemes
  - Phonemes: the sounds of language, corresponding roughly to letters
- Language model: phoneme strings to words and sentences
  - Statistical or rule-based (grammars)
  - Closed domain or open domain
- Early ASR used statistical acoustic models; around 2015 these were replaced by deep neural network models, improving performance significantly
- Grammar-based language models have been replaced by open domain statistical models, decreasing development effort

Natural Language Understanding (NLU)

- Extract relevant meaning from text
  - In many systems, meaning consists of "intents" (requested actions) and entities
  - In general in natural language, much more complex meanings can be conveyed: relations, negation, modality, counterfactuals, ...
- Until around 2000, NLU was mostly rule-based
  - A single grammar often used both to govern ASR and to extract meaning from text
- NLU is increasingly based on machine learning, generalising from examples

Dialogue Management (DM)

- Over the last 5-10 years there has been a focus in academia on statistical methods for dialogue management
- However, the complexity of dialogue management have lead to doubts about the prospects of such methods
- All commercial dialogue managers are more or less rule-based

Natural Language Generation (NLG)

- Convert output from DM into text
- NLG has so far received much less attention that ASR and NLU
- Many current commercial systems conflate DM and NLG, using simple language-templates with slot values filled in
  - "Calling $NAME's $NUMTYPE number"
- Research has produced more powerful generation techniques that are not being used commercially yet.
- Current approach works okay for simple kinds of dialogue and for syntactically simple languages such as English
- When moving into more complex domains and when localising to more complex languages (e.g. Turkish), NLG will become an issue

Text-To-Speech (TTS)

- TTS has improved significantly over the last 30 years, reaching almost natural voice quality
- However, there is still plenty of room for improvement
- For example, control over intonation is still a problem
- Example
  - "What **city** do you want to go to?"
  - "London"
  - # "What **city** do you want to go from?"

Text-To-Speech (TTS)

- TTS has improved significantly over the last 30 years, reaching almost natural voice quality
- However, there is still plenty of room for improvement
- For example, control over intonation is still a problem
- Example
  - "What **city** do you want to go to?"
  - "London"
  - "What city do you want to go **from**?"
- Generating correct intonation often requires some level of understanding of what is being said, and of what has been said before

Multimodality

- For practically useful dialogue systems, the connection between traditional touch-screen interaction and spoken interaction is important
- Current state of the art in industry is that the user has to choose between "normal" touch-screen interaction and spoken interaction (with a different GUI)
- Problems with this approach:
  - Forces users to abandon what they know for something less known
  - Not possible to mix spoken interaction and touch-screen interaction freely
  - Sometimes, you have to look at the screen
- Instead, systems should enable
  - The same touch-screen interaction regardless of whether speech is enabled or not
  - Users can switch modality anytime
  - Never necessary to look at the screen

Why is dialogue management important?

- Without a DM, there is no dialogue.
- The user has to give all information that the system needs in a single utterance, which in some cases may be very difficult and cognitively demanding
  - "I want to book a flight from Gothenburg to London on September 2 in the afternoon, coming back on the 10th in the morning, for 2 adults and 2 children aged 5 and 8, with no stopovers and preferably going to Heathrow airport, economy class."
- If any information is left out, there is no way to supply it later.

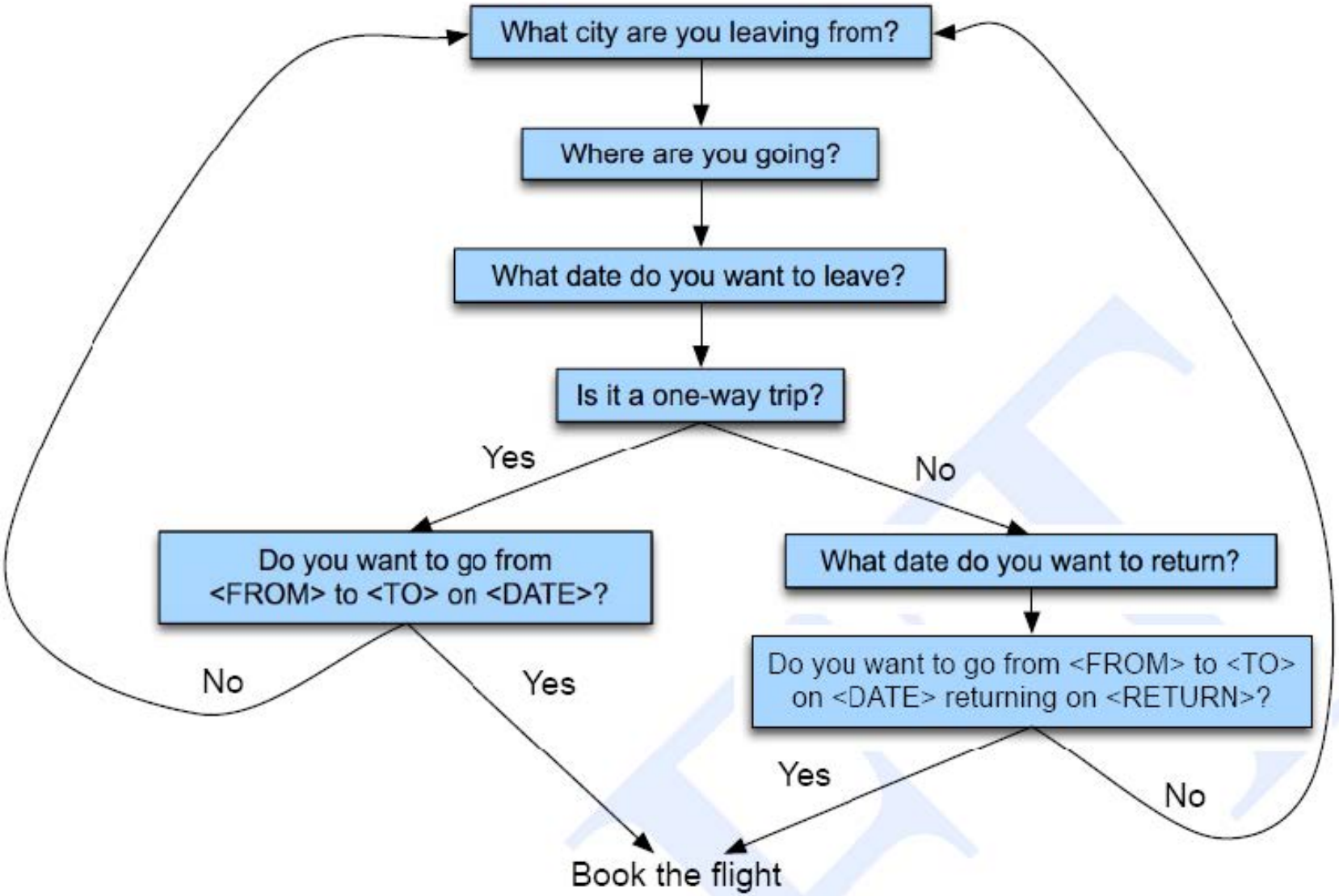
Why is dialogue management important?

- A dialogue manager makes it possible to have coherent exchanges consisting of several turns
- This means that the user does not have to say everything at once ("the truth, the whole truth and nothing but the truth")
- Instead, the user can say what's on her mind, and the system will ask for additional needed information

Dialogue Management methods

- Four types of dialogue managers:
  - Finite state-based
  - Form-filling
  - Plan-based
  - Information State

Finite state-based DM



Finite state-based DM

- Represents dialogue flow using a finite state machine
  - States: questions to the user
  - Transitions: user responses and resulting actions
  - Also stores answers in variables (<DATE> etc) (not pure finite state)
- Works for system initiative ("single initiative") dialogue
  - System has all the initiative
  - Tends to ignores or misinterpret anything which is not a direct answer to a system question

Finite state-based DM

- However, human-human conversation is very often "mixed initiative"
  - User may provide unrequested information
  - User may ask a question in response to a question
  - ...
- To deal with mixed initiative for n questions, $\sim 7n^2$ transitions are needed (for $n = 20$, 2800 states)
- These all need to be created and maintained by the dialogue developer

Form-based dialogue management

- Form = slots and values
- Relies on the structure of a form to guide the dialogue.
- Provides some aspects of mixed initiative dialogue
- Asks the user questions to fill slots in the frame
  - but allow the user to guide the dialogue by giving information that fills other slots in the frame
- Each slot may be associated with a question to ask the user, following type:
  - ORIGIN CITY "From what city are you leaving?"
  - DESTINATION CITY "Where are you going?"
  - DEPARTURE TIME "When would you like to leave?"
  - ARRIVAL TIME "When do you want to arrive?"

Form-based dialogue management

- DM asks questions to the user, filling any slot that the user specifies...
- ...until it has enough information to perform a data base query, and then return the result to the user
- If the user happens to answer two or three questions at a time, the system has to fill in these slots and then remember not to ask the user the associated questions for the slots.
- Does away with the strict constraints that the finite-state manager imposes

Form-based dialogue management

- VoiceXML
  - Voice Extensible Markup Language
  - an XML-based dialogue design language released by the W3C,
  - very simple mixed-initiative
  - form-based architecture
  - grammar-based ASR and NLU
- Most if not all systems on the market are more or less form-based (Siri, Google Assistant, etc.)
  - Statistical NLU has replaced grammars

Plan-based DM

- Popular 1980's-1990's
- View dialogue as planning and plan-recognition
- Highly general approach, can handle very complex dialogues (in principle)
- However:
  - Adapting such approaches to individual domains is very labour-intensive
  - Systems are very brittle and tend to break easily

Information State approach

- Goal: explore the space between finite-state/form-filling approaches (robust but limited) and plan-based approaches (capable but brittle and labour-intensive)
- Key component: a rich Information State, representing the state of the dialogue so far
- Deal with dialogue beyond form-filling in a robust way:
  - Dealing with multiple forms
  - Comparing alternatives ("negotiative dialogue")
  - General and versatile approaches to confirmation, turn-management and other basic dialogue phenomena
  - Instructional dialogue (e.g. technical manuals)
  - Problem-solving dialogue (e.g. putting together an itinerary)
- Important principle: "Separation of concerns"

Information State approach: separation of concerns

- Keep the following types of knowledge separate:
  - How to deal with the domain (domain knowledge)
  - How to speak about the domain (linguistic knowledge)
  - How to deal with dialogue (DM)
- Advantages
  - Simpler and faster development of new applications/domains, since only domain knowledge needs to be added
  - Simpler and faster localisation of applications to new languages, since only language knowledge needs to be added
  - Cumulative development of dialogue management since all DM improvements become available in future applications $\Rightarrow$ high quality DM across applications

Information State approach: Multiple forms

- Some domains require the ability to deal with multiple forms, e.g. for a travel agency application:
  - general route information ("Which airlines fly from Boston to San Francisco?")
  - information about airfare practices ("Do I have to stay a specific number of days to get a decent airfare?")
  - questions about car or hotel reservations
- Since users may want to switch between forms (in principle at any time), the system must be able to
  - disambiguate which slot of which form a given input is supposed to fill
  - switch dialogue control to that form
  - return control to previous form once the "embedded" form is done

	Industry	Academia
1990	Interactive Voice Response ▶ Finite state automata (FSA)	Rule-based systems ▶ Finite State-based, Form-filling, Plan-based DM ▶ Rule-based NLU ▶ Low quality ASR
2000	VoiceXML ▶ Finite-state-based, form-filling dialogue ▶ Rule-based NLU ▶ Grammar-based ASR	Information State Approach to DM ▶ Explore middle ground between form-filling and plan-based DM ▶ E.g. negotiative dialogue ▶ Separation of concerns
2010	Conversational assistants ▶ Form-filling dialogue ▶ Rule-based DM ▶ ASR gets a lot better	Machine learning approaches ▶ POMDP ▶ Reinforcement learning ▶ Back to form-filling dialogue ▶ Hardware advances for ML
2017	Development platforms ▶ Form-filling dialogue ▶ Rule-based DM ▶ ML for NLU, increased robustness	The pendulum swings back? ▶ Increased interaction with Industry ▶ Trend: need to move beyond form-filling

Outline

Machine learning vs. rule-based methods for dialogue systems

- Machine learning has proven useful for ASR and NLU, which are about extracting a meaningful message from a noisy signal
- Less useful for producing coherent responses (DM, NLG)
  - Machine learned methods are inherently unpredictable, but we often want the output from the system to be predictable (and debuggable)

Machine learning vs. rule-based methods for DM?

- Dialogue management has a huge state space compared to ASR and NLG, so a lot of (expensive) data is needed for machine learning
- Has proven very hard to get beyond form-based DM
- Keynotes at recent major conferences (SigDIAL, Interspeech) have made a case for revising rule-based DM and try to combine with ML, rather than trusting ML to solve everything

Outline

The future: Academia

- The pendulum is swinging back from purely ML approaches to DM, and there will be more work on hybrid approaches combining rule-based and ML methods for DM
- Theoretical work on human-human dialogue has made progress, and this needs to feed into DM research
- With more complex dialogue types comes higher demands on NLG and information presentation
- Work on robotics and dialogue will move towards embodied and situation-aware dialogue systems that can see what the user can see, and talk about it
- As systems become exposed to more diverse and less predictable environments, they will need to be able to learn language from users; foundational research is underway

The future: Industry

- Dialogue is coming into view, but has so far not received a lot of attention compared to ASR and NLU; this will eventually change
  - To some extent, dialogue can help with NLU problems, but this has yet to be exploited
- There will be a race to handle more complex types of dialogue
- Progress has been made on tools for building simple apps/skills; these need to be extended to work with more complex dialogue types
- For in-vehicle systems, managing cognitive load will be important
  - There is relevant academic research, e.g. about interrupting and resuming dialogue, and system-initiated dialogue