Dialogue Systems

고려대학교 컴퓨터학과
임희석 교수
limhseok@korea.ac.kr

What is Natural Language Dialogue?

- Communication involving
  - Multiple contributions
  - Coherent interaction
  - More than one participant
- Interaction modalities
  - Input: Speech, typing, writing, gesture
  - Output: Speech, text, graphical display, animated face/body (embodied virtual agent)

When is automatic dialogue system useful?

- When hands-free interaction is needed
  - In-car interface
  - In-field assistant system
  - Command-and-control interface
  - Language tutoring
  - Immersive training
- When speaking is easier than typing
  - Voice search interface
  - Virtual assistant (Siri, Google Now)
- Replacing human agents (cutting cost for companies)
  - Call routing
  - Menu-based customer help
  - Voice interface for customer assistance

What is involved in NL dialogue

- Understanding
  - What does a person say?
    - Identify words from speech signal
      - “Please close the window”
  - What does the speech mean?
    - Identify semantic content
      - Request (subject: close (object: window))
  - What were the speaker’s intentions?
    - Speaker requests an action in a physical world

What is involved in NL dialogue

- Managing interaction
  - Internal representation of the domain
  - Identify new information
  - Identifying which action to perform given new information
    - “close the window”, “set a thermostat” -> physical action
    - “what is the weather like outside?” -> call the weather API
  - Determining a response
    - “OK”, “I can’t do it”
    - Provide an answer
    - Ask a clarification question

What is involved in NL dialogue

- Access to information
- To process a request “Please close the window” you (or the system) needs to know:
  - There is a window
  - Window is currently opened
  - Window can/can not be closed

What is involved in NL dialogue

- Producing language
  - Deciding when to speak
  - Deciding what to say
    - Choosing the appropriate meaning
  - Deciding how to present information
    - So partner understands it
    - So expression seems natural

Types of dialogue systems (1/2)

- Command and control
  - Actions in the world
  - Robot - situated interaction
- Information access
  - Database access
    - Bus/train/airline information
    - Librarian
    - Voice manipulation of a personal calendar
  - API access

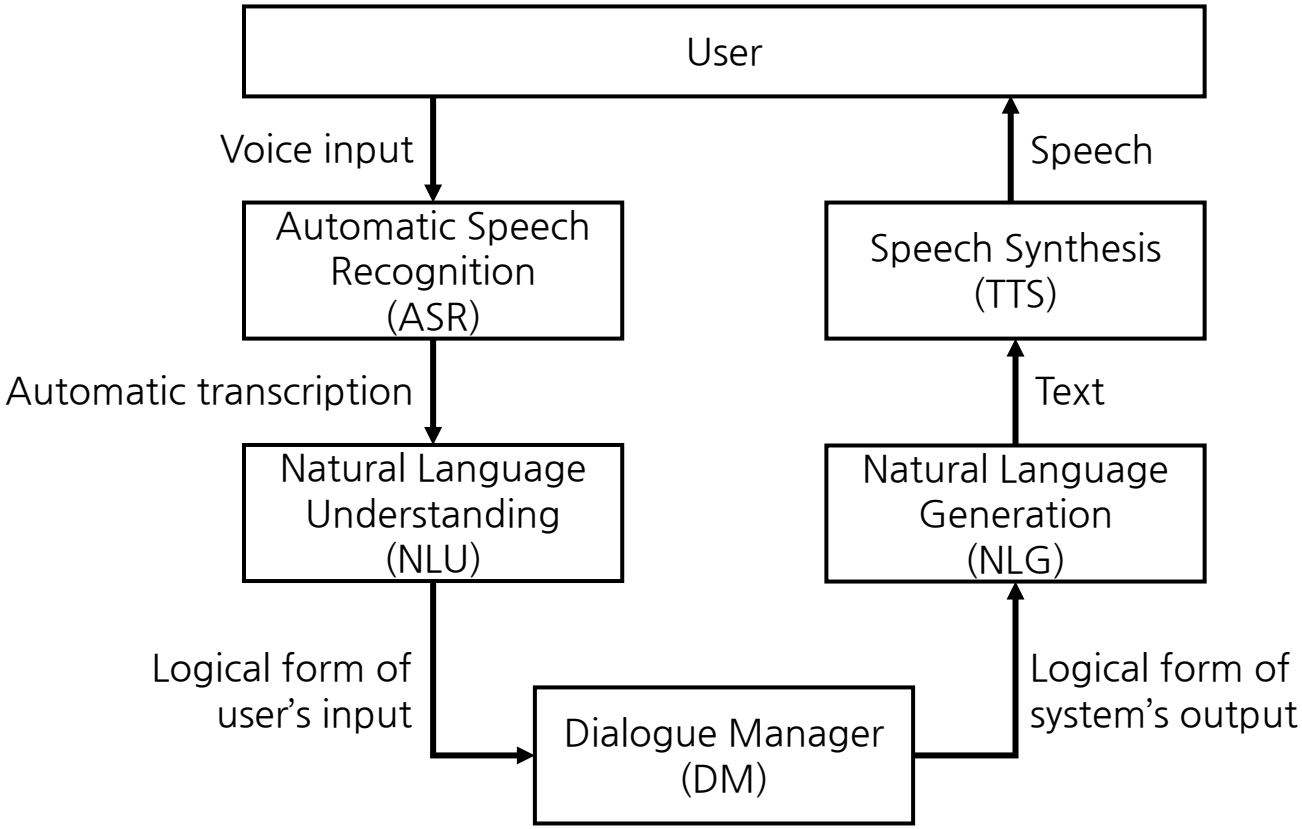
Types of dialogue systems (2/2)

- Customer service
  - Simple call routing
  - Menu-based interaction
  - Allows flexible response “How may I help you?”
- Smart virtual assistant (vision)
  - Helps you perform tasks, such as buying movie tickets, trouble shooting
  - Reminds you about important events without explicit reminder settings

Aspects of Dialogue Systems

- Which modalities does the system use
  - Voice only (telephone/microphone & speaker)
  - Voice and graphics (smartphones)
  - Virtual human
    - Can show emotions
  - Physical device
    - Can perform actions
- Back-end
  - which resources (database/API/ontology) it accesses
- How much world knowledge does the system have
  - Hand-built ontologies
  - Automatically learned from the web
- How much personal knowledge does it have and use
  - Your calendar (google)
  - Where you live/work (google)
  - Who are your friends/relatives (facebook)

Dialogue System Components



Speech Recognition

Speech recognition

- ASR is the problem of automatically transcribing captured audio samples of human speech
  - Input: an audio sample
  - Output: sequence of words

Use of ASR in Dialogue Systems

- Selection of a speech recognizer
- Resources that may be needed
  - Grammar
  - Language model
  - Phonetic dictionary
  - Acoustic model
- Tool for capturing audio

Grammar-based

- Appropriate when you can easily circumscribe the syntactic form of user utterances
- Grammars are typically hand-crafted for the domain
  - For example, if your system expects digits a rule:
  - S -> zero | one | two | three | …
  - Advantages: better performance on in-domain speech
  - Disadvantages: does not recognize out-of-domain
- Grammar format may vary by ASR
  - E.g. Sphinx takes JSGF format grammars (regular language, see http://www.w3.org/TR/jsgf/)

Open Domain

- Use language models(LM) to assign probabilities to word sequences
- Statistical LMs are trained using a collection of sample utterances
  - Large text samples are preferable
- Generally used when the syntactic form of user utterances is hard to circumscribe
- Advantages: can potentially recognize any word sequence
- Disadvantages: lower performance on in-domain utterances (digits may be misrecognized)

Acoustic Models

- An acoustic model computes the probability of the observed acoustic features in an audio sample given a word (phone) sequence
- May be trained on 5-10 hours of speech (or much more)
- Sphinx provides US English acoustic models for
  - microphone and broadcast speech
  - telephone speech
- Default acoustic models may work less well for different recording environments or accented English (UK English, Indian English)

ASR Output Options

- 1-best text hypothesis
  - Most common approach in implemented dialogue systems
- N-best text hypotheses or word lattice
  - May offer advantages in certain domains
  - See De Mori et al (2008)

Some readily available ASRs

- Research systems (highly configurable)
  - Kaldi
    - Most used research recognizer
  - Sphinx/pocket sphinx
    - Open source ASR
    - Java API
- Industry (free cloud version), not configurable
  - Google Speech API
  - Nuance
  - AT&T Watson

Speech recognition

- Challenges: recognition errors due to
  - Noisy environment
  - Speaker accent
  - Speaker interruption, self correction, etc.

SYS:	Welcome to the CMU Let's Go bus information. What can I do for you?
USR: ASR:	I need to go from Oakland *I need to go .from. can't*
SYS:	Where do you wanna leave from?
USR: ASR:	to Waterfront *told. .me. Waterfront*
SYS:	Leaving from Waterfront. Did I get that right?
USR: *ASR:*	Oakland *Oakland*
	…

Natural Language Understanding

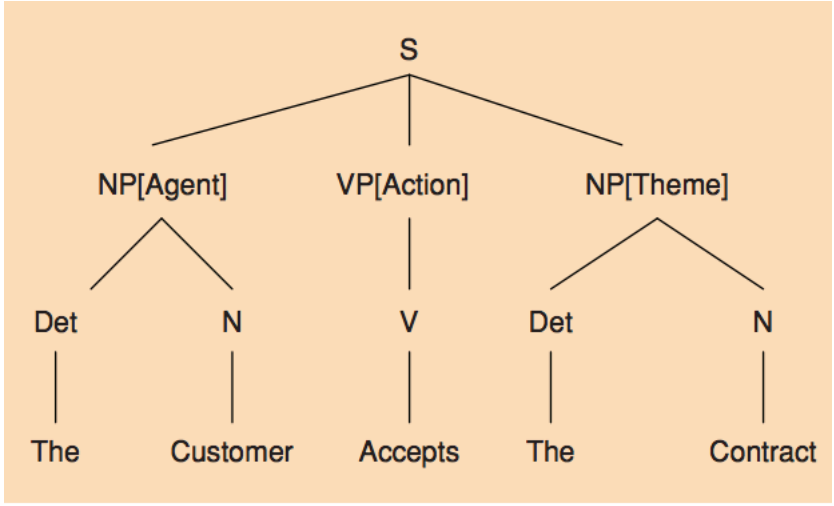
Natural Language Understanding

- Convert input text into internal representation. Example internal representation in wit.ai:


```
{
"msg_body": "what is playing at Lincoln Center",
 "outcome": {
  "intent": "get_shows",
  "entities": {
   "Venue": {
    "value": "Lincoln Center",
   }
  },
  "confidence": 0.545
 },
 "msg_id": "c942ad0f-0b63-415f-b1ef-84fbfa6268f2"
}
```

NLU approaches

- Can be based on simple phrase matching
  - “leaving from PLACE”
  - “arriving at TIME”
- Can use deep or shallow syntactic parsing



NLU approaches

- Can be rule-based
  - Rules define how to extract semantics from a string/syntactic tree
- Or Statistical
  - Train statistical models on annotated data
    - Classify intent
    - Tag named entities

Possible Inputs to a NLU Module

- ASR output (1-best, N-best, or lattice)
- Dialogue context of the utterance
  - Simple summary of state of dialogue
    - State in a finite state model of the dialogue interaction
    - Key aspects of dialogue history (as in Phoenix)
    - Last system utterance
  - Information state representation of dialogue state
    - Can encode arbitrary aspects of dialogue history
- Other knowledge resources (e.g. database)

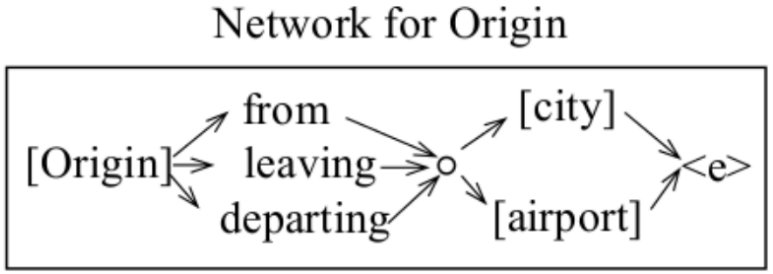
Possible Outputs from a NLU Module

- Different dialogue systems formalize the NLU problem in different ways
- Some common NLU outputs include:
  - Slot values
  - Frames
  - Speech act labels
  - Speech act label + semantic content

NLU Output: Slot Values

- Slot values can be identified with pattern matching directly on the text input to NLU

User: … from Denver ...



- Slot-matching patterns may be regular expressions (FSAs) or context-free grammars (RTNs)
- Some words may be skipped between matched slots, improving robustness
- Slot-values may be combined with more complex NLU outputs to capture details like numeric values

NLU output: Frames

- A frame is a collection of slot-values
  - Very flexible representation
  - Can decompose the meaning of an utterance into the components that are meaningful to a dialogue system
  - Can have hierarchical structure
  - Values can be shared across slots
  - The slot-value framework can be used to encode various kinds of semantic representations
- Frame outputs can be constructed in many ways
  - Slot-value parsing (as in the Phoenix parser)
  - Data-driven statistical classification (as in mxNLU)
  - Syntactic parsing + semantic rules

NLU Output: Speech Act Labels

- Speech acts capture aspects of utterances
  - Arose from a theoretical view of spoken utterances as actions (Austin, 1962; Searle 1969)
- Taxonomies of speech act types may be defined
  - Greeting, acknowledging, asserting, offering, etc.
- Speech act types are often used to represent what type of action a user utterance is making from the system's perspective

Speech Act Taxonomy Example

- Switchboard SWBD-DAMSL (Jurafsky, Shriberg, and Biasca, 1997)
  - http://groups.inf.ed.ac.uk/switchboard/dialactmanual.html

SWBD-DAMSL	Example	Count	%
Statement-non-opinion	Me, I'm in the legal department.	72,824	36%
Acknowledge (Backchannel)	Uh-huh.	37,096	19%
Statement-opinion	I think it's great.	25,197	13%
Agree/Accept	That's exactly it.	10,820	5%
Abandoned or Turn-Exit	So, -	10,569	5%
Yes-No-Question	Do you have to have any special training?	4,624	2%
Non-verbal	[Laughter], [Throat_clearing]	3,548	2%
…			

Other Speech Act Taxonomies

- Similar terms: dialogue acts, dialogue moves, conversation acts, etc.
- Further references:
  - Bunt et al. (2010), Towards an ISO standard for dialogue act annotation.
  - Traum (2000), 20 Questions for Dialogue Act Taxonomies, in Journal of Semantics, 17(1):7-30

NLU Output: Speech Act Label + Semantic Content

- Dialogue systems generally need to know more than the type of speech act
- They also need the content of that speech act

User: *we are prepared to give you guys generators for electricity*

NLU output:

$$\begin{bmatrix} mood : declarative \\ sem : \begin{bmatrix} type : event \\ agent : captain-kirk \\ event : deliver \\ theme : power-generator \\ source : us-army \\ modal : [\, possibility : can \,] \\ speech-act : [\, type : offer \,] \end{bmatrix} \end{bmatrix}$$

Other Approaches to Frame-based NLU

- Can use a syntactic parser + semantic rules
  - See e.g. De Mori et al. (2008)
- Can use a tagging model (e.g. CRF) to tag individual words in the word sequence with frame elements (slot values)
  - See e.g. Heintze et al. (2010)
- Can build an ensemble of classifiers for each slot
  - See e.g. Heintze et al. (2010)
- Many other approaches possible (e.g. MT-based)

Dialogue Manager

Dialogue Manager (DM)

- Is a “brain” of an SDS
- Decides on the next system action/dialogue contribution
- SDS module concerned with dialogue modeling
  - Dialogue modeling: formal characterization of dialogue, evolving context, and possible/likely continuations

DM approaches

- Rule-based
  - Key phrase reactive
  - Finite state/Tree based
    - model the dialogue as a path through a tree or finite state graph structure
  - Information-state Update
- Statistical (learn state transition rules from data or on-line)
- Hybrid (a combination of rules and statistical method)

Dialogue Management

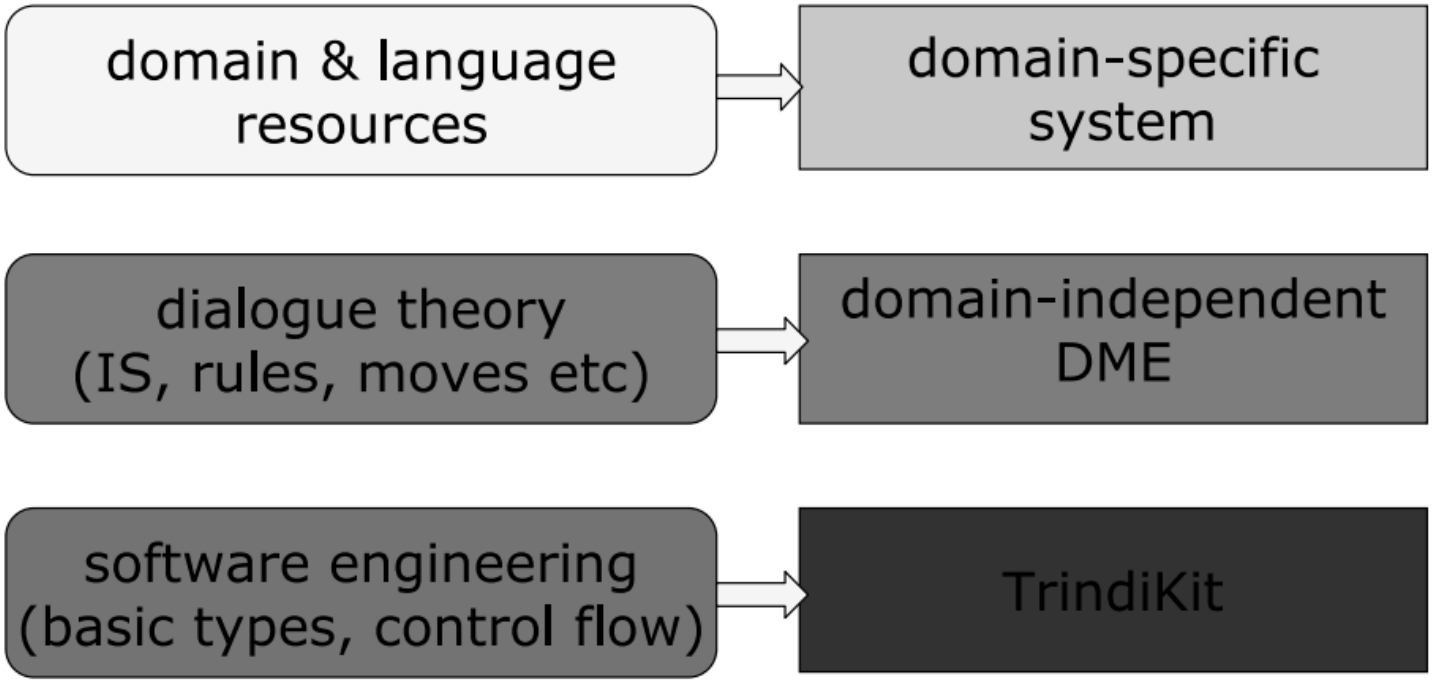
- Address how SDS is implemented
  - Reusability of components

- Approaches:
  - Frame-based approach (structural)
  - Information State approach (Traum & Larsson 2003)

Information State Approach

- Formalizes theories of dialogue
  - Speech act
- Combines different theoretical approaches to SDS
  - Planning (more flexible and complex)
  - Structural (simple scripted dialogues)
- Leads to better engineering of SDS components
  - Separate development of modules
  - Facilitates reuse of components

ISU Architecture



Information State DM

- Formalize DM function as Information State Update
  - Identify relevant aspects of information
    - How are they updated
    - What controls this updates
- Dialogue context
  - Current state of the system including
    - Known information
- Role of DM
  - Update dialogue context
  - Provide context-dependent expectations
  - Interface with task/domain processing
  - Decide what context to express next

Information state

- Private:
  - Belief set
  - Agenda: stack of actions
- Shared:
  - Belief set
  - QUD: stack of questions
  - LM: move

Natural Language Generation

NLG approaches

- Presenting semantic content to the user
- Template-based
  - In a airline reservation system:
  - User: “Find me a ticket from New York to London”
  - System: “What date do you want to travel?”
  - User: “March 10”
  - System: “There is a United flight from Newark airport to London Heathrow on March 10 leaving at 9:15 AM”
    - Template: There is a AIRLINE flight from AIRPORT to AIRPORT on DATE leaving at TIME

Natural language generation (NLG)

- Content selection
  - User asks “Find me restaurants in Chelsea”
  - System finds 100 restaurants
  - NLG decides how to present a response and which information to present
    - “I found 100 restaurants, the restaurant with highest rating is …”
    - “I found 100 restaurants, the closest to you is …”
    - “I found 100 restaurants, I think you would like …”

실습

실습

- 본인이 원하는 하나의 도메인(날씨, 길찾기 제외)을 정하여 wit.ai를 이용한 Dialog System 구현
- 클라이언트 언어: 자유
- 제출 내용: 클라이언트 소스 코드, 보고서
- 보고서 내용: 시스템의 목적, 구현 방법 및 알고리즘, 시스템이 수용 가능한 대화 예시, 실행 화면 스크린샷, 본인의 wit.ai 앱 주소 (https://wit.ai/[wit.ai_아이디]/[wit.ai_앱이름])

채점 항목

- 보고서
  - 목적, 구현 방법, 대화 예시, 실행 화면
- Wit.ai
  - Wit.ai 사용 여부
- 소스 코드
  - 작동 여부, wit.ai api 호출, 현실 데이터 사용 여부