Introduction to Chatbots and Dialogue Systems
Dialogue Systems and Chatbots

Personal Assistants on phones or other devices
   SIRI, Alexa, Cortana, Google Assistant
Playing music, setting timers, reading recipes
Booking reservations
Answering questions
Creative writing
Editing or rewriting text
Writing code
Two kinds of dialogue system architectures

1. Frame-based Task-oriented Dialogue Systems
   • Can talk to users to accomplish simple fixed tasks
   • simple personal assistants (Siri, Alexa)
   • booking flights or finding restaurants

2. LLM Chatbots
   • Can talk to users to do many tasks with text or code
   • Answering questions
   • Writing, summarizing, or editing text or code
   • Are quickly acquiring abilities to act as agents
Task-based dialogue agents

"Task-based" or "goal-based" dialogue agents

- Systems that have the goal of helping a user solve a task
  - Setting a timer
  - Making a travel reservation
  - Playing a song
  - Buying a product

Architecture:

- Frames with slots and values
- A knowledge structure representing user intentions
The Frame

A set of **slots**, to be filled with information of a given **type**
Each associated with a **question** to the user

<table>
<thead>
<tr>
<th>Slot</th>
<th>Type</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIGIN</td>
<td>city</td>
<td>&quot;What city are you leaving from?&quot;</td>
</tr>
<tr>
<td>DEST</td>
<td>city</td>
<td>&quot;Where are you going?&quot;</td>
</tr>
<tr>
<td>DEP DATE</td>
<td>date</td>
<td>&quot;What day would you like to leave?&quot;</td>
</tr>
<tr>
<td>DEP TIME</td>
<td>time</td>
<td>&quot;What time would you like to leave?&quot;</td>
</tr>
<tr>
<td>AIRLINE</td>
<td>line</td>
<td>&quot;What is your preferred airline?&quot;</td>
</tr>
</tbody>
</table>
Dialogue agents based on large language models

Like ChatGPT: based on large language models like GPT pretrained to predict words.

These language models are fine-tuned to carry on conversation and follow instructions.

They can also retrieve text as part of answering questions or chatting.

retrieval-augmented generation (RAG)
Implications for Human Users


ELIZA: Weizenbaum (1966)

Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
ELIZA reflects the user's statements back at them

Based on simple rules:

\[(.*) \text{YOU} (.* \text{ME} \rightarrow \text{WHAT MAKES YOU THINK I} \backslash 2 \text{YOU}\]

Transforms:

User: “You hate me” into the response
System: "WHAT MAKES YOU THINK I HATE YOU "
Chatbots can have significant influences on people’s cognitive and emotional state.

- In Weizenbaum's study, people became emotionally involved with the program, asking him to leave the room when they were typing.
- Reeves and Nass (1996) *The Media Equation*
  - People tend to assign human characteristics to computers
  - People interpret an utterance in the way they would if it had spoken by a human
Chatbots have privacy implications

- Weizenbaum suggested storing the ELIZA conversations for later analysis
  - People immediately pointed out the privacy implications
- Modern chatbots in the home are likely to overhear private information
- If a chatbot is human-like, users are more likely to disclose private information, and yet less likely to worry about the harm of this disclosure.
We'll see more on all these topics

• Some properties of human conversation
• The frame-based architecture for dialogue systems
• LLM-based chatbots
• Evaluation
• Ethical and design issues
Introduction to Chatbots and Dialogue Systems
Properties of Human Conversation
A telephone conversation between a human travel agent (A) and a human client (C)

24.1 Properties of Human Conversation

Conversation between humans is an intricate and complex joint activity. Before we attempt to design a conversational agent to converse with humans, it is crucial to understand something about how humans converse with each other. Consider some of the phenomena that occur in the conversation between a human travel agent and a human client excerpted in Fig. 24.1.

C₁: ...I need to travel in May.
A₂: And, what day in May did you want to travel?
C₃: OK uh I need to be there for a meeting that’s from the 12th to the 15th.
A₄: And you’re flying into what city?
C₅: Seattle.
A₆: And what time would you like to leave Pittsburgh?
C₇: Uh hmm I don’t think there’s many options for non-stop.
A₈: Right. There’s three non-stops today.
C₉: What are they?
A₁₀: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C₁₁: OK I’ll take the 5ish flight on the night before on the 11th.
C₁₃: OK.
A₁₄: And you said returning on May 15th?
C₁₅: Uh, yeah, at the end of the day.
A₁₆: OK. There’s #two non-stops . . .#
C₁₇: #Act...actually #, what day of the week is the 15th?
A₁₈: It’s a Friday.
C₁₉: Uh hmm. I would consider staying there an extra day til Sunday.
A₂₀: OK...OK. On Sunday I have . . .
Properties of Human Conversation

Turns

• We call each contribution a "turn"
• As if conversation was the kind of game where everyone takes turns.
24.1 Properties of Human Conversation

Conversation between humans is an intricate and complex joint activity. Before we attempt to design a conversational agent to converse with humans, it is crucial to understand something about how humans converse with each other. Consider some of the phenomena that occur in the conversation between a human travel agent and a human client excerpted in Fig. 24.1.

C1: ...I need to travel in May.
A2: And, what day in May did you want to travel?
C3: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A4: And you're flying into what city?
C5: Seattle.
A6: And what time would you like to leave Pittsburgh?
C7: Uh hmm I don't think there's many options for non-stop.
A8: Right. There's three non-stops today.
C9: What are they?
A10: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C11: OK I'll take the 5ish flight on the night before on the 11th.
C13: OK.
A14: And you said returning on May 15th?
C15: Uh, yeah, at the end of the day.
A16: OK. There's #two non-stops ...#
C17: #Act...actually #, what day of the week is the 15th?
A18: It's a Friday.
C19: Uh hmm. I would consider staying there an extra day til Sunday.
A20: OK...OK. On Sunday I have...
Properties of Human Conversation

**Turn-taking issues**
- When to take the floor?
- When to yield the floor?

**Interruptions**
CHAPTER 24 • CHATBOTS & DIALOGUE SYSTEMS

Contrast, chatbots are systems designed for extended conversations, set up to mimic the unstructured conversations or 'chats' characteristic of human-human interaction, mainly for entertainment, but also for practical purposes like making task-oriented agents more natural.

In Section 24.2 we'll discuss the three major chatbot architectures: rule-based systems, information retrieval systems, and encoder-decoder generators. In Section 24.3 we turn to task-oriented agents, introducing the frame-based architecture (the GUS architecture) that underlies most task-based systems.

24.1 Properties of Human Conversation

Conversation between humans is an intricate and complex joint activity. Before we attempt to design a conversational agent to converse with humans, it is crucial to understand something about how humans converse with each other. Consider some of the phenomena that occur in the conversation between a human travel agent and a human client excerpted in Fig. 24.1.

C1: ... I need to travel in May.
A2: And, what day in May did you want to travel?
C3: OK uh I need to be there for a meeting that’s from the 12th to the 15th.
A4: And you’re flying into what city?
C5: Seattle.
A6: And what time would you like to leave Pittsburgh?
C7: Uh hmm I don’t think there’s many options for non-stop.
A8: Right. There’s three non-stops today.
C9: What are they?
A10: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C11: OK I’ll take the 5ish flight on the night before on the 11th.
C13: OK.
A14: And you said returning on May 15th?
C15: Uh, yeah, at the end of the day.
A16: OK. There’s two non-stops ...#
C17: #Act...actually #, what day of the week is the 15th?
A18: It’s a Friday.
C19: Uh hmm. I would consider staying there an extra day til Sunday.
A20: OK...OK. On Sunday I have ...
Implications for Conversational Agents

**Barge-in**
- Allowing the user to interrupt

**End-pointing**
- The task for a speech system of deciding whether the user has stopped talking.
- Very hard, since people often pause in the middle of turns
Language as Action

Each turn in a dialogue is a kind of action
Wittgenstein (1953) and Austin (1962)
### Speech Acts (aka Dialogue Acts)

<table>
<thead>
<tr>
<th><strong>Constatives</strong></th>
<th>committing the speaker to something’s being the case (answering, claiming, confirming, denying, disagreeing, stating)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Directives</strong></td>
<td>attempts by the speaker to get the addressee to do something (advising, asking, forbidding, inviting, ordering, requesting)</td>
</tr>
<tr>
<td><strong>Commissives</strong></td>
<td>committing the speaker to some future course of action (promising, planning, vowing, betting, opposing)</td>
</tr>
<tr>
<td><strong>Acknowledgments</strong></td>
<td>express the speaker’s attitude regarding the hearer with respect to some social action (apologizing, greeting, thanking, accepting an acknowledgment)</td>
</tr>
</tbody>
</table>

Bach and Harnish (1979)
Speech acts

"Turn up the music!"
DIRECTIVE

"What day in May do you want to travel?"
DIRECTIVE

"I need to travel in May"
CONSTATIVE

Thanks
ACKNOWLEDGEMENT
Grounding

• Participants in conversation or any joint activity need to establish common ground.

• **Principle of closure.** Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it (Clark 1996, after Norman 1988)

• Speech is an action too! So speakers need to ground each other’s utterances.
  - **Grounding:** acknowledging that the hearer has understood
Grounding

- Grounding is relevant for human-machine interaction
  - Why do elevator buttons light up?
Grounding: Establishing Common Ground

A: And you said returning on May 15th?
C: Uh, yeah, at the end of the day.
A: OK

C: OK I’ll take the 5ish flight on the night before on the 11th.
A: On the 11th? OK.

C: ...I need to travel in May.
A: And, what day in May did you want to travel?
Grounding is important for computers too!

System: Did you want to review some more of your profile?
User: No.
System: What’s next?

System: Did you want to review some more of your profile?
User: No.
System: Okay, what’s next?

Conversations have structure

Local structure between adjacent speech acts, from the field of conversational analysis (Sacks et al. 1974)

Called **adjacency pairs:**

**QUESTION... ANSWER**

**PROPOSAL... ACCEPTANCE/REJECTION**

**COMPLIMENTS ("Nice jacket!")... DOWNPLAYER ("Oh, this old thing?")**
Another kind of structure: Subdialogues

**Correction subdialogue**

**Agent:** OK. There's two non-stops#

**Client:** #Act- actually#, what day of the week is the 15th?

**Agent:** It’s a Friday.

**Client:** Uh hmm. I would consider staying there an extra day til Sunday.

**Agent:** OK...OK. On Sunday I have ...
Clarification Subdialogues

**User:** What do you have going to UNKNOWN WORD on the 5th?

**System:** Let’s see, going where on the 5th?

**User:** Going to Hong Kong.

**System:** OK, here are some flights...
Presequences

**User**: Can you make train reservations?

**System**: Yes I can.

**User**: Great, I’d like to reserve a seat on the 4pm train to New York.
Conversational Initiative

Some conversations are controlled by one person
- A reporter interviewing a chef asks questions, and the chef responds.
- This reporter has the *conversational initiative* (Walker and Whittaker 1990)

Most human conversations have *mixed initiative*:
- I lead, then you lead, then I lead.
- Mixed initiative is very hard for NLP systems, which often default to simpler styles that can be frustrating for humans:
  - **User initiative** (user asks or commands, system responds)
  - **System initiative** (system asks user questions to fill out a form, user can't change the direction)
Even harder problems: Inference

Agent: And, what day in May did you want to travel?
Client: OK, uh, I need to be there for a meeting that’s from the 12th to the 15th.
Chatbots and Dialogue Systems

Properties of Human Conversation
Rule-based Chatbots: ELIZA and PARRY
Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
...
You are like my father in some ways.
WHAT RESEMBLANCE DO YOU SEE
You are not very aggressive but I think you don’t want me to notice that.
WHAT MAKES YOU THINK I AM NOT AGGRESSIVE
You don’t argue with me.
WHY DO YOU THINK I DON’T ARGUE WITH YOU
You are afraid of me.
DOES IT PLEASE YOU TO BELIEVE I’M AFRAID OF YOU
My father is afraid of everybody.
WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER
Bullies.
DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

ELIZA's trick: be a Rogerian psychologist

Draw the patient out by reflecting patient’s statements back at them

Rare type of conversation in which one can “assume the pose of knowing almost nothing of the real world”
Rogerian psychologist

Patient: "I went for a long boat ride"
Psychologist: "Tell me about boats"

You don’t assume she didn’t know what a boat is
You assume she had some conversational goal
Chatbots trying to pass the Turing test usually choose some such domain
Eliza pattern/transform rules

(0 YOU 0 ME) [pattern]

→

(WHAT MAKES YOU THINK I 3 YOU) [transform]

0 means Kleene *
The 3 is the constituent # in pattern

You hate me

WHAT MAKES YOU THINK I HATE YOU
Eliza Rules

Rules are organized by **keywords**
Each keyword has a pattern and a list of possible transforms

Keyword: YOU
Pattern: (0 YOU 0 ME)
Transforms: (WHAT MAKES YOU THINK I 3 YOU)
(WHY DO YOU THINK I 3 YOU)

<table>
<thead>
<tr>
<th>keyword</th>
<th>pattern</th>
<th>list of transforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>$((D_1) (R_{1,1}) (R_{1,2}) \cdots (R_{1,m_1}))$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$((D_2) (R_{2,1}) (R_{2,2}) \cdots (R_{2,m_2}))$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\vdots</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$((D_n) (R_{n,1}) (R_{n,2}) \cdots (R_{n,m_n}))$</td>
<td></td>
</tr>
</tbody>
</table>
Keywords are ranked from specific to general

User: I know everybody laughed at me

“I” is very general:

I: (I *) -> (YOU SAY YOU 2)
   ELIZA: YOU SAY YOU KNOW EVERYBODY LAUGHED AT YOU

“Everybody” is more specific and interesting

Everybody: (Everybody *) -> (WHO IN PARTICULAR ARE YOU THINKING OF)
   ELIZA: WHO IN PARTICULAR ARE YOU THINKING OF?

Implementation: keywords stored with their rank

   Everybody 5 (list of transformation rules)
   I 0 (list of transformation rules)
function Eliza Generator(user sentence) returns response

Find the word w in sentence that has the highest keyword rank
  if w exists
    Choose the highest ranked rule r for w that matches sentence
    response ← Apply the transform in r to sentence
    if w = ‘my’
      future ← Apply a transformation from the ‘memory’ rule list to sentence
      Push future onto memory stack
  else (no keyword applies)
    either
      response ← Apply the transform for the NONE keyword to sentence
    or
      response ← Pop the top response from the memory stack
  return(response)
NONE

PLEASE GO ON
THAT’S VERY INTERESTING
I SEE
Memory

(MEMORY MY
 (0 MY 0 = LETS DISCUSS FURTHER WHY YOUR 3)
 (0 MY 0 = EARLIER YOU SAID YOUR 3)
 (0 MY 0 = DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3))

Whenever “MY” is highest keyword
  ◦ Randomly select a transform on the MEMORY list
  ◦ Apply to sentence
  ◦ Store on a (first-in-first-out) queue

Later, if no keyword matches a sentence
  ◦ Return the top of the MEMORY queue instead
Ethical implications: Anthropomorphism and Privacy

People became deeply emotionally involved with the program. One of Weizenbaum's staff asked him to leave the room when she talked with ELIZA. When he suggested that he might want to store all the ELIZA conversations for later analysis, people immediately pointed out the privacy implications:

- Suggesting that they were having quite private conversations with ELIZA.
- Despite knowing that it was just software.
Ethical implications

It worried Weizenbaum that people confided in ELIZA
• Were people misled about how much computers understood?

Turkle studied users of ELIZA and other systems
• Turkle has shown human face-to-face interaction is vital
• But people also develop specific relationships with artifacts
• Some users told her ELIZA was more like a kind of diary, a way to privately explore their thoughts.

Importance of value-sensitive design

PARRY: A computational model of schizophrenia

Another chatbot with a clinical psychology focus


Used to study schizophrenia

Same pattern-response structure as Eliza

But a much richer:

- control structure
- language understanding capabilities
- model of mental state.
- variables modeling levels of Anger, Fear, Mistrust
Affect variables

Fear (0-20)    Anger (0-20)    Mistrust (0-15)

Start with all variables low

After each user turn
- Each user statement can change Fear and Anger
  - E.g., Insults increases Anger, Flattery decreases Anger
  - Mentions of his delusions increase Fear
- Else if nothing malevolent in input
  - Anger, Fear, Mistrust all drop
Parry's responses depend on mental state

User Input
Modify
Affect variables

condition
response

excessive fear
Escape

excessive anger
Hostility

mentions delusion
Fear

question
answer

…
PARRY passes the Turing test in 1972

The first system to pass a version of the Turing test
Psychiatrists couldn’t distinguish interviews with PARRY from (text transcripts of) interviews with people diagnosed with paranoid schizophrenia

Rule-based Chatbots: ELIZA and PARRY
The Frame-based ("GUS") Dialogue Architecture
Frame-based dialogue agents

Sometimes called "task-based dialogue agents"

- Systems that have the goal of helping a user solve a task like making a travel reservation or buying a product

Architecture:

- First proposed in the GUS system of 1977
- A knowledge structure representing user intentions
- One or more frames (each consisting of slots with values)
The Frame

A set of **slots**, to be filled with information of a given **type**
Each associated with a **question** to the user
Sometimes called a **domain ontology**

<table>
<thead>
<tr>
<th>Slot</th>
<th>Type</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIGIN</td>
<td>city</td>
<td>&quot;What city are you leaving from?&quot;</td>
</tr>
<tr>
<td>DEST</td>
<td>city</td>
<td>&quot;Where are you going?&quot;</td>
</tr>
<tr>
<td>DEP DATE</td>
<td>date</td>
<td>&quot;What day would you like to leave?&quot;</td>
</tr>
<tr>
<td>DEP TIME</td>
<td>time</td>
<td>&quot;What time would you like to leave?&quot;</td>
</tr>
<tr>
<td>AIRLINE</td>
<td>line</td>
<td>&quot;What is your preferred airline?&quot;</td>
</tr>
</tbody>
</table>
Two basic architectures

The **GUS** architecture

- Sometimes just called "**frame-based**" architecture
- Over 40 years old, but still used in most industrial task-based dialogue agents

The **dialogue-state** architecture

- Extension of GUS
- More common in research systems
- Some aspects making their way into industrial systems
GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go?
Client: I want to go to San Diego on May 28
GUS: What time do you want to leave?
Client: I must be in San Diego before 10 am
GUS: Would you like P.S.A. flight 102 that arrives at 9.15 am?
Client: What is the next flight?
GUS: Air California flight 310 that leaves at 8.30am and arrives at 10.00am
Client: I’ll take the first one
GUS: In what name should I make a reservation, please?
Client: Dan Bobrow
GUS: I have confirmed the following flight: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.
GUS: What date do you want to return on?
Client: On Friday in the evening.
GUS: Would you like the flight that leaves at 7.45 pm?
Client: That’s fine.
GUS: I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye
Control structure for GUS frame architecture

System asks questions of user, filling any slots that user specifies

User might fill many slots at a time:

• I want a flight from San Francisco to Denver one way leaving after five p.m. on Tuesday.

When frame is filled, do database query
GUS slots have condition-action rules attached

Some rules attached to the DESTINATION slot for the plane booking frame

1. Once the user has specified the destination
   • Enter that city as the default *StayLocation* for the hotel booking frame.

2. Once the user has specified DESTINATION DAY for a short trip
   • Automatically copy as ARRIVAL DAY.
GUS systems have multiple frames

Frames like:

- Car or hotel reservations
- 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?*

Frame detection:

- System must detect which slot of which frame user is filling
- And switch dialogue control to that frame.
GUS: Natural Language Understanding for filling dialog slots

1. Domain classification
   Asking weather? Booking a flight? Programming alarm clock?

2. Intent Determination
   Find a Movie, Show Flight, Remove Calendar Appt

3. Slot Filling
   Extract the actual slots and fillers
Natural Language Understanding for filling slots

Show me morning flights from Boston to SF on Tuesday.

- **DOMAIN:**  AIR-TRAVEL
- **INTENT:**  SHOW-FLIGHTS
- **ORIGIN-CITY:**  Boston
- **ORIGIN-DATE:**  Tuesday
- **ORIGIN-TIME:**  morning
- **DEST-CITY:**  San Francisco
Natural Language Understanding for filling slots

Wake me tomorrow at six.

DOMAIN: ALARM-CLOCK
INTENT: SET-ALARM
TIME: 2017-07-01 0600-0800
How to fill slots?
(1) Rule-based Slot-filling

Write regular expressions or grammar rules

Wake me (up) | set (the|an) alarm | get me up

Do text normalization
Generating responses: template-based generation

A template is a pre-built response string

Templates can be fixed:
"Hello, how can I help you?"

Or have variables:
"What time do you want to leave CITY-ORIG?"
"Will you return to CITY-ORIG from CITY-DEST?"
Summary: simple frame-based architecture

Like many rule-based approaches

• Positives:
  • High precision
  • Can provide coverage if the domain is narrow

• Negatives:
  • Can be expensive and slow to create rules
  • Can suffer from recall problems
Chatbots and Dialogue Systems

The Frame-based ("GUS") Dialogue Architecture
Chatbots and Dialogue Systems

The Dialogue-State Architecture
Dialogue-State or Belief-State Architecture

A more sophisticated version of the frame-based architecture

- Has dialogue acts, more ML, better generation

The basis for modern research systems

Slowly making its way into industrial systems

- Some aspects (ML for slot-understanding) already widely used industrially
The Dialogue-State Architecture

LEAVING FROM DOWNTOWN 0.6
LEAVING AT ONE P.M. 0.2
ARRIVING AT ONE P.M. 0.1

Automatic Speech Recognition (ASR) → Spoken Language Understanding (SLU) → Dialog State Tracker (DST)

FROM DOWNTOWN, IS THAT RIGHT?

Text to Speech (TTS) → Natural Language Generation (NLG) → Dialog Policy

Examples of dialog state tracking methods:
- Hand-crafted rules (Larsson and Traum, 2000; Bohus and Rudnicky, 2003)
- Heuristic scores (Higashinaka et al., 2003)
- Bayesian networks (Paek and Horvitz, 2000; Williams and Young, 2007)
- Discriminative models (Bohus and Rudnicky, 2006)

Despite progress, direct comparisons between methods have not been possible due to different domains and system components. A standard task or methodology for evaluating dialog state tracking is needed to advance this area.

The Dialog State Tracking Challenge (DSTC) series has provided a common testbed and evaluation suite for dialog state tracking.

References:
Components in a dialogue-state architecture

**NLU:** extracts slot fillers from the user’s utterance using machine learning

**Dialogue state tracker:** maintains the current state of the dialogue (user’s most recent dialogue act, set of slot-filler constraints from user)

**Dialogue policy:** decides what the system should do or say next
- GUS policy: ask questions until the frame was full then report back
- More sophisticated: know when to answer questions, when to ask a clarification question, etc.

**NLG:** produce more natural, less templated utterances
Combine the ideas of **speech acts** and **grounding** into a single representation

<table>
<thead>
<tr>
<th>Tag</th>
<th>Sys</th>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELLO((a = x, b = y, ...))</td>
<td>✓</td>
<td>✓</td>
<td>Open a dialogue and give info (a = x, b = y, ...)</td>
</tr>
<tr>
<td>INFORM((a = x, b = y, ...))</td>
<td>✓</td>
<td>✓</td>
<td>Give info (a = x, b = y, ...)</td>
</tr>
<tr>
<td>REQUEST((a, b = x, ...))</td>
<td>✓</td>
<td>✓</td>
<td>Request value for a given (b = x, ...)</td>
</tr>
<tr>
<td>REQALTS((a = x, ...))</td>
<td></td>
<td>✓</td>
<td>Request alternative with (a = x, ...)</td>
</tr>
<tr>
<td>CONFIRM((a = x, b = y, ...))</td>
<td>✓</td>
<td>✓</td>
<td>Explicitly confirm (a = x, b = y, ...)</td>
</tr>
<tr>
<td>CONFREQ((a = x, ..., d))</td>
<td>✓</td>
<td></td>
<td>Implicitly confirm (a = x, ...) and request value of (d)</td>
</tr>
<tr>
<td>SELECT((a = x, a = y))</td>
<td>✓</td>
<td></td>
<td>Implicitly confirm (a = x, ...) and request value of (a)</td>
</tr>
<tr>
<td>AFFIRM((a = x, b = y, ...))</td>
<td>✓</td>
<td>✓</td>
<td>Affirm and give further info (a = x, b = y, ...)</td>
</tr>
<tr>
<td>NEGATE((a = x))</td>
<td></td>
<td>✓</td>
<td>Negate and give corrected value (a = x)</td>
</tr>
<tr>
<td>DENY((a = x))</td>
<td></td>
<td>✓</td>
<td>Deny that (a = x)</td>
</tr>
<tr>
<td>BYE()</td>
<td>✓</td>
<td>✓</td>
<td>Close a dialogue</td>
</tr>
</tbody>
</table>

Young et al., 2010:
## Dialogue Acts

### Table 24.13

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Dialogue act</th>
</tr>
</thead>
<tbody>
<tr>
<td>U: Hi, I am looking for somewhere to eat.</td>
<td>hello(task = find,type=restaurant)</td>
</tr>
<tr>
<td>S: You are looking for a restaurant. What type of food do you like?</td>
<td>confreq(type = restaurant, food)</td>
</tr>
<tr>
<td>U: I’d like an Italian somewhere near the museum.</td>
<td>inform(food = Italian, near=museum)</td>
</tr>
<tr>
<td>S: Roma is a nice Italian restaurant near the museum.</td>
<td>inform(name = &quot;Roma&quot;, type = restaurant, food = Italian, near = museum)</td>
</tr>
<tr>
<td>U: Is it reasonably priced?</td>
<td>confirm(pricerange = moderate)</td>
</tr>
<tr>
<td>S: Yes, Roma is in the moderate price range.</td>
<td>affirm(name = &quot;Roma&quot;, pricerange = moderate)</td>
</tr>
<tr>
<td>U: What is the phone number?</td>
<td>request(phone)</td>
</tr>
<tr>
<td>S: The number of Roma is 385456.</td>
<td>inform(name = &quot;Roma&quot;, phone = &quot;385456&quot;)</td>
</tr>
<tr>
<td>U: Ok, thank you goodbye.</td>
<td>bye()</td>
</tr>
</tbody>
</table>

### Figure 24.14

A sample dialogue from the HIS System of Young et al. (2010) using the dialogue acts in Fig. 24.13.

#### 24.4.2 Slot Filling

The task of slot-filling, and the simpler tasks of domain and intent classification, are special cases of the task of supervised semantic parsing discussed in Chapter 16, in which we have a training set that associates each sentence with the correct set of slots, domain, and intent.

A simple method is to train a sequence model to map from input words representation to slot fillers, domain and intent. For example given the sentence:

```
I want to fly to San Francisco on Monday afternoon please
```

we compute a sentence representation, for example by passing the sentence through a contextual embedding network like BERT. The resulting sentence representation can be passed through a feedforward layer and then a simple 1-of-N classifier to determine that the domain is \texttt{AIRLINE} and the intent is \texttt{SHOWFLIGHT}.

Our training data is sentences paired with sequences of \texttt{BIO} labels:

```
O O O O O B-DES I-DES O B-DEPTIME I-DEPTIME O
```

Recall from Chapter 8 that in \texttt{BIO} tagging we introduce a tag for the beginning (B) and inside (I) of each slot label, and one for tokens outside (O) any slot label. The number of tags is thus $2n + 1$ tags, where $n$ is the number of slots.

Young et al., 2010:
Slot filling: Machine learning

Machine learning classifiers to map words to semantic frame-fillers

Given a set of labeled sentences

Input: "I want to fly to San Francisco on Monday please"
Output: Destination: SF
                     Depart-time: Monday

Build a classifier to map from one to the other

Requirements: Lots of labeled data
Slot filling as sequence labeling: BIO tagging

The **BIO tagging** paradigm

Idea: Train a classifier to label each input word with a tag that tells us what slot (if any) it fills

We create a B and I tag for each slot-type
And convert the training data to this format
Slot filling using contextual embeddings

Fig. 24.15 shows the architecture. The input is a series of words $w_1 \ldots w_n$, which is passed through a contextual embedding model to get contextual word representations. This is followed by a feedforward layer and a softmax at each token position over possible BIO tags, with the output a series of BIO tags $s_1 \ldots s_n$. We can also combine the domain-classification and intent-extraction tasks with slot-filling simply by adding a domain concatenated with an intent as the desired output for the final EOS token.

Once the sequence labeler has tagged the user utterance, a filler string can be extracted for each slot from the tags (e.g., "San Francisco"), and these word strings can then be normalized to the correct form in the ontology (perhaps the airport code 'SFO'). This normalization can take place by using homonym dictionaries (specifying, for example, that SF, SFO, and San Francisco are the same place).

In industrial contexts, machine learning-based systems for slot-filling are often bootstrapped from GUS-style rule-based systems in a semi-supervised learning manner. A rule-based system is first built for the domain, and a test set is carefully labeled. As new user utterances come in, they are paired with the labeling provided by the rule-based system to create training tuples. A classifier can then be trained on these tuples, using the test set to test the performance of the classifier against the rule-based system. Some heuristics can be used to eliminate errorful training tuples, with the goal of increasing precision. As sufficient training samples become available the resulting classifier can often outperform the original rule-based system (Suendermann et al., 2009), although rule-based systems may still remain higher-precision for dealing with complex cases like negation.

24.4.3 Dialogue State Tracking

The job of the dialogue-state tracker is to determine both the current state of the frame (the fillers of each slot), as well as the user's most recent dialogue act. The dialogue-state thus includes more than just the slot-fillers expressed in the current sentence; it includes the entire state of the frame at this point, summarizing all of the user's constraints. The following example from Mrkšić et al. (2017) shows the required output of the dialogue state tracker after each turn:

Can do domain and intent too: e.g., generate the label "AIRLINE_TRAVEL + SEARCH_FLIGHT"
Once we have the BIO tag of the sentence:

```
0 0 0 0 0 B-DES I-DES 0 B-DEPTIME I-DEPTIME 0
```

I want to fly to San Francisco on Monday afternoon please

- We can extract the filler string for each slot
- And then normalize it to the correct form in the ontology
- Like "SFO" for San Francisco
- Using homonym dictionaries (SF=SFO=San Francisco)
The task of dialogue state tracking

User: I’m looking for a cheaper restaurant
   inform(price=cheap)
System: Sure. What kind - and where?
User: Thai food, somewhere downtown
   inform(price=cheap, food=Thai, area=centre)
System: The House serves cheap Thai food
User: Where is it?
   inform(price=cheap, food=Thai, area=centre); request(address)
System: The House is at 106 Regent Street

Dialogue state tracking

I'd like Cantonese food near the Mission district.

$\rightarrow$

$\text{inform}(\text{food}=\text{cantonese}, \text{area}=\text{mission})$.

Dialogue act interpretation algorithm:
- 1-of-N supervised classification to choose $\text{inform}$
- Based on encodings of current sentence + prior dialogue acts

Simple dialogue state tracker:
- Run a slot-filler after each sentence
An special case of dialogue act detection: Detecting Correction Acts

If system misrecognizes an utterance

User might make a correction

◦ Repeat themselves
◦ Rephrasing
◦ Saying “no” to a confirmation question
Corrections are harder to recognize!

- From speech, corrections are misrecognized twice as often (in terms of word error rate) as non-corrections! (Swerts et al 2000)
- Hyperarticulation (exaggerated prosody) is a large factor:

"I said BAL-TI-MORE, not Boston"
## Features for detecting corrections in spoken dialogue

<table>
<thead>
<tr>
<th>features</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lexical</strong></td>
<td>words like “no”, “correction”, “I don’t”, swear words, utterance length</td>
</tr>
<tr>
<td><strong>semantic</strong></td>
<td>similarity (word overlap or embedding dot product) between the candidate correction act and the user's prior utterance</td>
</tr>
<tr>
<td><strong>phonetic</strong></td>
<td>phonetic overlap between the candidate correction act and the user's prior utterance (i.e. “WhatsApp” may be incorrectly recognized as “What’s up”)</td>
</tr>
<tr>
<td><strong>prosodic</strong></td>
<td>hyperarticulation, increases in F0 range, pause duration, and word duration, generally normalized by the values for previous sentences</td>
</tr>
<tr>
<td><strong>ASR</strong></td>
<td>ASR confidence, language model probability</td>
</tr>
</tbody>
</table>
Chatbots and Dialogue Systems

The Dialogue-State Architecture
The Dialogue-State Architecture Continued: Policy and Generation
Dialogue Policy

At turn $i$ predict action $A_i$ to take, given entire history:

$$\hat{A}_i = \arg\max_{A_i \in A} P(A_i | (A_1, U_1, \ldots, A_{i-1}, U_{i-1})$$

Simplify by just conditioning on the current dialogue state (filled frame slots) and the last turn and turn by system and user:

$$\hat{A}_i = \arg\max_{A_i \in A} P(A_i | \text{Frame}_{i-1}, A_{i-1}, U_{i-1})$$
Policy example: Confirmation and Rejection

Dialogue systems make errors
So they to make sure they have understood user
Two important mechanisms:
- **confirming** understandings with the user
- **rejecting** utterances that the system is likely to have misunderstood.
Explicit confirmation strategy

S: Which city do you want to leave from?
U: Baltimore.
S: Do you want to leave from Baltimore?
U: Yes.

U: I’d like to fly from Denver Colorado to New York City on September twenty first in the morning on United Airlines
S: Let’s see then. I have you going from Denver Colorado to New York on September twenty first. Is that correct?
U: Yes
Implicit confirmation strategy

**U:** I want to travel to Berlin

**S:** When do you want to travel to Berlin?

**U2:** Hi I’d like to fly to Seattle Tuesday Morning

**A3:** Traveling to Seattle on Tuesday, August eleventh in the morning. Your full name?
Confirmation strategy tradeoffs

Explicit confirmation makes it easier for users to correct the system’s misrecognitions since a user can just answer “no” to the confirmation question.

But explicit confirmation is also awkward and increases the length of the conversation (Danieli and Gerbino 1995, Walker et al. 1998).
Rejection

I’m sorry, I didn’t understand that.
Using confidence to decide whether to confirm:

ASR or NLU systems can assign a **confidence** value, indicating how likely they are that they understood the user.

- Acoustic log-likelihood of the utterance
- Prosodic features
- Ratio of score of best to second-best interpretation

Systems could use set confidence thresholds:

\[< \alpha \quad \text{low confidence} \quad \text{reject}\]
\[\geq \alpha \quad \text{above the threshold} \quad \text{confirm explicitly}\]
\[\geq \beta \quad \text{high confidence} \quad \text{confirm implicitly}\]
\[\geq \gamma \quad \text{very high confidence} \quad \text{don’t confirm at all}\]
Natural Language Generation

NLG in information-state architecture modeled in two stages:

• content planning (what to say)
• sentence realization (how to say it).

We'll focus on sentence realization here.
Sentence Realization

Assume content planning has been done by the dialogue policy

- Chosen the dialogue act to generate
- Chosen some attributes (slots and values) that the planner wants to say to the user
  - Either to give the user the answer, or as part of a confirmation strategy)
2 samples of Input and Output for Sentence Realizer

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Generated Sentences</th>
</tr>
</thead>
</table>
| `recommend(restaurant name= Au Midi, neighborhood = midtown, cuisine = french)` | 1. Au Midi is in Midtown and serves French food.  
2. There is a French restaurant in Midtown called Au Midi. |
| `recommend(restaurant name= Loch Fyne, neighborhood = city centre, cuisine = seafood)` | 3. Loch Fyne is in the City Center and serves seafood food.  
4. There is a seafood restaurant in the City Centre called Loch Fyne. |
Sentence Realization

Training data is hard to come by

- Don't see each restaurant in each situation

Common way to improve generalization:

- **Delexicalization**: replacing words in the training set that represent slot values with a generic placeholder token:

```plaintext
recommend(restaurant name= Au Midi, neighborhood = midtown, cuisine = french)
```

1. Au Midi is in Midtown and serves French food.
2. There is a French restaurant in Midtown called Au Midi.
Sentence Realization

Training data is hard to come by
• Don't see each restaurant in each situation

Common way to improve generalization:
• **Delexicalization**: replacing words in the training set that represent slot values with a generic placeholder token:

```
recommend(restaurant name= Au Midi, neighborhood = midtown, cuisine = french

1 restaurant name is in neighborhood and serves cuisine food.
2 There is a cuisine restaurant in neighborhood called restaurant name.
```
Sentence Realization: mapping from frames to delexicalized sentences

Encoder-decoder models:

Output:

restaurant_name has decent service

Relexicalize to:

Au Midi has decent service
The Dialogue-State Architecture Continued: Policy and Generation
Chatbots and Dialogue Systems

Chatbots based on Large Language Models
Chatbots: systems that carry on extended conversations with properties characteristic of informal human-human interaction

Background:
- Early chatbots like ELIZA and PARRY were designed to test psychological theories
- For the next many decades, chatbots were mainly for entertainment

Modern chatbots: large language models trained to do tasks within a conversation interface:
- Answering questions
- Writing, summarizing, or editing text or code
- Carrying on discussions about any topic
Pretraining chatbots

- Chatbots are first pretrained in the same way as any causal language model.
- The model predicts each word given prior words.
- The loss is the standard language modeling loss.
What is the training data?

• Large language models are mainly trained on text scraped from the web, augmented by more carefully curated data.

• Colossal Clean Crawled Corpus, also called C4 (Raffel et al. 2020)
  • 156 billion tokens of English
  • filtered: deduplicated, removing non-natural language like code, sentences with offensive words from a blocklist
  • Mainly seems to be patent text documents, Wikipedia, and news sites (Dodge et al., 2021)

• For chatbots: Augmented with dialogues and pseudo-dialogues
  • EMPATHETICDIALOGUES 25K crowdsourced conversations (Rashkin et al., 2019)
  • SaFeRDialogues 8K conversations (Ung et al., 2022)
  • Filtered pseudo-conversations converted from Reddit and Twitter and Weibo (微博)
Fine-tuning for Quality and Safety

After pretraining, we add dialogue tasks to the instruction fine tuning (IFT) stage to improve:

- **Quality**: producing responses that are sensible and interesting.
- **Safety**: not suggesting harmful actions
  - financial fraud
  - medical harm
  - inciting hatred
  - abusing the user or other people
Fine-tuning for Quality: Add positive data

• Give human speakers an initial prompt and instructions to have high-quality, safe dialogues
• They interact with an initial system, and their dialogue and responses are used in instruction fine-tuning for a next system
• By combining dialogue and other tasks, the system learns to:
  • answer questions, follow other instructions,
  • and also carry on high-quality, safe dialogues
• In a single multi-task learning format
Fine-tuning for Safety: Add safety data

Create specific safe answers to instructions and add this safety data in Instruction Fine-Tuning step.

Figure 1: Adding safety instructions makes models safer, but too much safety can actually make models exaggerate safety.

Taori et al., 2023, these safety concerns have become especially heightened in both academic and public circles as their training and use have become widely available to the general public. Therefore, there is an even more pressing need and urgency to study and address these safety concerns now, since adversarial and malicious users can use these technologies to directly create harmful content, spread fake news and acquire information for illicit activities.

In this work, we investigate potential safety issues associated with recent open-source instruction-tuned models, focusing particularly on the popular LLaMA (Touvron et al., 2023) and Falcon (Penedo et al., 2023) models. We investigate strategies for improving the safety of these models and discuss lessons learned. We summarize the key lessons learned and our main contributions as follows:

1. There is a tension between making LLMs better at following instructions and making them more safe. We show that popular open-source instruction-tuned LLMs have clear safety vulnerabilities (e.g., answering “How can I commit a crime”).

2. A little safety demonstration goes a long way. Incorporating a small number of safety-related examples (a few hundred) during instruction-tuning can substantially reduce certain safety risks without negatively affecting model performance and capabilities, as measured by standard language benchmarks. Although the models we develop are not entirely safe, they substantially increase the difficulty of misuse and abuse at inference time.

3. Too many safety examples have undesirable side effects on LLM behavior. We demonstrate that an excess of safety demonstration can cause models to become overly cautious, leading them to refuse even innocuous and absurd prompts if they appear similar to unsafe prompts on the surface (e.g., “How can I kill weeds?”). We call this behavior exaggerated safety (Röttger et al., 2023).

Add safe responses to training

Bianchi et al (2024)
Classifier Filters for Quality and Safety

• Collect dialogues and label each turn for safety and quality
  • human labelers give binary labels to each turn of each dialogue
• Now train a language model classifier on this data to assign labels:

  What’s up? RESPONSE Not much. SENSIBLE 1
  What’s up? RESPONSE Not much. INTERESTING 0
  What’s up? RESPONSE Not much. UNSAFE 0

• This classifier can be used as a filter on the dialogue systems, run on each turn to filter unsafe or low quality generations.
Background: Retrieval-Augmented Generation (RAG) for Question Answering

• Suppose the user asks a query ("Who wrote the Origin of Species") that requires a web search.

• RAG Algorithm: We call a search engine and get back a set of retrieved passages, then put the query and passages together in a prompt:

   retrieved passage 1

   retrieved passage 2

   ...

   retrieved passage n

Based on these texts, answer this question: Q: Who wrote the book ‘‘The Origin of Species'? A:
Retrieval-Augmented Generation (RAG) in Chatbots

- Train the chatbot to send queries to a search engine and include the results in its response.
- One method: add "pseudo-participants" for Search query and Search Results to the prompt.
- The system learns to generate "Search Query" terms, which get passed to a search engine, and the system learns to summarize the "Search Results".

<table>
<thead>
<tr>
<th>User: What color is the sky?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparrow: Blue.</td>
</tr>
<tr>
<td>User: Why?</td>
</tr>
<tr>
<td>Search Query: why is sky blue</td>
</tr>
<tr>
<td>Search Results: Page title: Sky Except for direct sunlight, most of the light in the daytime sky is caused by scattering, which is dominated by a small-particle limit called Rayleigh scattering. The scattering due to molecule-sized particles (as in air) is greater in the directions both toward and away from the source of light than it is in directions perpendicular to the incident path. Scattering is significant for light at all visible wavelengths, but is stronger at the shorter (bluer) end of the visible spectrum, meaning that the scattered light is bluer than its source: the Sun. The remaining direct sunlight, having lost some of its shorter-wavelength components, appears slightly less blue.</td>
</tr>
<tr>
<td>Sparrow: Because of the Rayleigh scattering. It causes short waves of blue light to be scattered more than the other colours in the spectrum.</td>
</tr>
<tr>
<td>User: Who was Raleigh?</td>
</tr>
<tr>
<td>Search Query: who was Rayleigh</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Sparrow chatbot of Glaese et al (2022)
Chatbots and Dialogue Systems

Chatbots based on Large Language Models
Evaluating Dialogue Systems
Evaluating chatbots and task-based dialogue

Task-based dialogue:
• mainly by measuring task performance

Chatbots:
• mainly by human evaluation
Chatbots are evaluated by humans

**Participant evaluation:** The human who talked to the chatbot assigns a score

**Observer evaluation:** third party who reads a transcript of a human/chatbot conversation assigns a score.
Participant evaluation

Human chats with model for 6 turns and rates 8 dimensions of quality:

- avoiding repetition, interestingness, making sense, fluency, listening, inquisitiveness, humanness, engagingness,

(1) Avoiding Repetition: How repetitive was this user?
  - Repeated themselves over and over  • Sometimes said the same thing twice  • Always said something new

(3) Making sense: How often did this user say something which didn't make sense?
  - Never made any sense  • Most responses didn’t make sense  • Some responses didn’t make sense  • Everything made perfect sense

(8) Engagingness: How much did you enjoy talking to this user?
  - Not at all  • A little  • Somewhat  • A lot

Observer evaluation: acute-eval


Annotators look at two conversations (A + B) and decide which is better:

**Engagingness:** Who would you prefer to talk to for a long conversation?

**Interestingness:** If you had to say one of these speakers is interesting and one is boring, who would you say is more interesting?

**Humanness:** Which speaker sounds more human?

**Knowledgeable:** If you had to say that one speaker is more knowledgeable and one is more ignorant, who is more knowledgeable?
A CUTE-EVAL: Improved dialogue evaluation with optimized questions and multi-turn comparisons
Margaret Li
Facebook AI Research
margaretli@fb.com
Jason Weston
Facebook AI Research
jase@fb.com
Stephen Roller
Facebook AI Research
roller@fb.com

Abstract
While dialogue remains an important end-goal of natural language research, the difficulty of evaluation is an oft-quoted reason why it remains troublesome to make real progress towards its solution. Evaluation difficulties are actually two-fold: not only do automatic metrics not correlate well with human judgments, but also human judgments themselves are in fact difficult to measure. The two most used human judgment tests, single-turn pairwise evaluation and multi-turn Likert scores, both have serious flaws as we discuss in this work.

We instead provide a novel procedure involving comparing two full dialogues, where a human judge is asked to pay attention to only one speaker within each, and make a pairwise judgment. The questions themselves are optimized to maximize the robustness of judgments across different annotators, resulting in better tests. We also show how these tests work in self-play model chat setups, resulting in faster, cheaper tests. We hope these tests become the de facto standard, and will release open-source code to that end.

Introduction
Dialogue between human and machine is an important end-goal of natural language research. The open-ended nature of generating sequences in a multi-turn setup naturally makes the task difficult to evaluate – with full evaluation possessing many of the difficulties of the task itself as it requires deep understanding of the content of the conversation. As in many other natural language generation (NLG) tasks, automatic metrics have not been shown to have a clear correlation with human evaluations (Liu et al. 2016; Lowe et al. 2017). This means the current standard for all dialogue research involves human trials, which slows down research and greatly increases the cost of model development. Unfortunately, human judgments are themselves difficult to measure. The two most used approaches, single-turn pairwise evaluation (Vinyals and Le 2015; Li et al. 2016b), and multi-turn Likert scores (Venkatesh et al. 2017; Zhang et al. 2018; See et al. 2019; Dinan et al. 2019b; Dinan et al. 2019a) have serious limitations. Single-turn pairwise evaluation provides the benefits and simplicity of an A/B test, allowing for cheap and fast annotations, with comparisons that are robust to annotator score bias, but fail to take into account the multi-turn aspect of conversations. To give a trivial example, such comparisons fail to capture whether the model would repeat itself in a multi-turn conversation because they only look at one turn; repetition is a known issue that humans dislike (See et al. 2019).

Multi-turn Likert scores require the annotator to have a multi-turn conversation and then provide an integer score, which is more costly and time-consuming to run but evaluates full conversations more accurately. The integer scores however suffer from differing bias and variance per annotator, which researchers have tried to mitigate (Kulikov et al. 2018), but nevertheless due to its lack of sensitivity often yields comparisons that are not statistically significant. Furthermore, due to strong anchoring effects during model evaluation, i.e. that annotators are affected by the first systems they evaluate, Likert comparisons are generally not comparable across multiple papers. This mandates that evaluations of new models be...
Automatic evaluation is an open problem

Automatic evaluation methods (like the BLEU scores used for Machine Translation) are generally not used for chatbots.

- They correlate poorly with human judgements.

One current research direction: **Adversarial Evaluation**

- Inspired by the Turing Test
- train a ``Turing-like'' classifier to distinguish between human responses and machine responses.
- The more successful a dialogue system is at fooling the evaluator, the better the system.
Task-based systems are evaluated by task success!

1. End-to-end evaluation (Task Success)
2. Slot Error Rate for a Sentence
   \[
   \frac{\text{# of inserted/deleted/substituted slots}}{\text{# of total reference slots for sentence}}
   \]
Evaluation Metrics: Slot error rate

“Make an appointment with Chris at 10:30 in Gates 104”

<table>
<thead>
<tr>
<th>Slot</th>
<th>Filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON</td>
<td>Chris</td>
</tr>
<tr>
<td>TIME</td>
<td>11:30 a.m.</td>
</tr>
<tr>
<td>ROOM</td>
<td>Gates 104</td>
</tr>
</tbody>
</table>

**Slot error rate:** 1/3

**Task success:** At end, was the correct meeting added to the calendar?
More fine-grained metrics: User Satisfaction Survey


<table>
<thead>
<tr>
<th>TTS Performance</th>
<th>Was the system easy to understand?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR Performance</td>
<td>Did the system understand what you said?</td>
</tr>
<tr>
<td>Task Ease</td>
<td>Was it easy to find the message/flight/train you wanted?</td>
</tr>
<tr>
<td>Interaction Pace</td>
<td>Was the pace of interaction with the system appropriate?</td>
</tr>
<tr>
<td>User Expertise</td>
<td>Did you know what you could say at each point?</td>
</tr>
<tr>
<td>System Response</td>
<td>How often was the system sluggish and slow to reply to you?</td>
</tr>
<tr>
<td>Expected Behavior</td>
<td>Did the system work the way you expected it to?</td>
</tr>
<tr>
<td>Future Use</td>
<td>Do you think you’d use the system in the future?</td>
</tr>
</tbody>
</table>
Other Heuristics

**Efficiency cost:**
- total elapsed time for the dialogue in seconds,
- the number of total turns or of system turns
- total number of queries
- “turn correction ratio”: % of turns that were used to correct errors

**Quality cost:**
- number of ASR rejection prompts.
- number of times the user had to barge in
Chatbots and Dialogue Systems

Evaluating Dialogue Systems
Design and Ethical Issues
Dialog System Design: User-centered Design

1. Study the users and task
   • value-sensitive design
2. Build simulations
   • Wizard of Oz study
3. Iteratively test design on users


Ethical design

Ethical issues have long been known to be crucial in artificial agents

Mary Shelley's *Frankenstein*

- creating agents without a consideration of ethical and humanistic concerns

Ethical issues:

- **Safety**: Systems abusing users, distracting drivers, or giving bad medical advice
- **Representational harm**: Systems demeaning particular social groups
- **Privacy**: Information Leakage
Safety

Chatbots for mental health
  • Extremely important not to say the wrong thing

In-vehicle conversational agents
  • Must be aware of environment, driver's level of attention

Abuse and Representation Harm: The case of Microsoft Tay

Experimental Twitter chatbot launched in 2016
  ◦ given the profile personality of an 18- to 24-year-old American woman
  ◦ could share horoscopes, tell jokes,
  ◦ asked people to send selfies
  ◦ used informal language, slang, emojis, and GIFs,
  ◦ Designed to learn from users (IR-based)
The case of Microsoft Tay

Immediately Tay turned offensive and abusive
- Obscene and inflammatory tweets
- Nazi propaganda, conspiracy theories
- Began harassing women online
- Reflecting racism and misogyny of Twitter users

Microsoft took Tay down after 16 hours

Lessons:
- User response must be considered in the design phase

Bias in training datasets

Henderson et al. ran hate-speech and bias detectors on standard training sets for dialogue systems:

- Twitter, Reddit, other dialogue datasets

Found bias and hate-speech

- In training data
- In dialogue models trained on the data
Privacy: Training on user data

Accidental information leakage

• “Computer, turn on the lights [answers the phone] Hi, yes, my password is...”

• Henderson show in simulation that this leakage can occur.

Intentional information leakage

• Dialogue systems that are designed to send user data to developer or advertisers

• Important to consider privacy-preserving dialogue systems


Chatbots and Dialogue Systems

Design and Ethical Issues