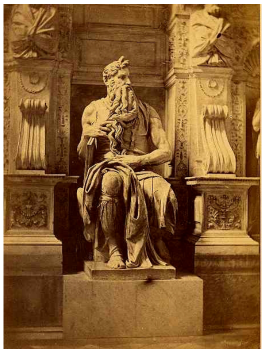
CHAPTER

15 Chatbots & Dialogue Systems

> Les lois de la conversation sont en général de ne s'y appesantir sur aucun objet, mais de passer légèrement, sans effort et sans affectation, d'un sujet à un autre ; de savoir y parler de choses frivoles comme de choses sérieuses
>
> [The rules of conversation are, in general, not to dwell on any one subject, but to pass lightly from one to another without effort and without affectation; to know how to speak about trivial topics as well as serious ones;]
>
> The 18th C. *Encyclopedia* of Diderot, start of the entry on conversation

conversation
dialogue

The literature of the fantastic abounds in inanimate objects magically endowed with the gift of speech. From Ovid's statue of Pygmalion to Mary Shelley's story about Frankenstein, we continually reinvent stories about creating something and then having a chat with it. Legend has it that after finishing his sculpture *Moses*, Michelangelo thought it so lifelike that he tapped it on the knee and commanded it to speak. Perhaps this shouldn't be surprising. Language is the mark of humanity and sentience, and **conversation** or **dialogue** is the most fundamental arena of language. It is the first kind of language we learn as children, and the kind we engage in constantly, whether we are ordering lunch, buying train tickets, or talking with our families, friends, or coworkers.



dialogue system
frame
chatbot

This chapter introduces the fundamental algorithms of programs that use conversation to interact with users. We often distinguish between two kinds of architectures. **Task-oriented dialogue systems** converse with users to accomplish fixed tasks like controlling appliances or finding restaurants, relying on a data structure called the **frame**, which represents the knowledge a system needs to acquire from the user (like the time to set an alarm clock). **Chatbots**, by contrast, are designed to mimic the longer and more unstructured conversations or 'chats' characteristic of human-human interaction. Modern systems incorporate aspects of both; industrial chatbots like ChatGPT can carry on longer unstructured conversations; industrial digital assistants like Siri or Alexa are generally frame-based dialogue systems.

The fact that chatbots and dialogue systems are designed for human-computer interaction has strong implications for their design and use. Many of these implications already became clear in one of the earliest chatbots, ELIZA (Weizenbaum, 1966). ELIZA was designed to simulate a Rogerian psychologist, based on a branch of clinical psychology whose methods involve drawing the patient out by reflecting patient's statements back at them. Rogerian interactions are the rare type of conversation in which, as Weizenbaum points out, one can "assume the pose of knowing almost nothing of the real world". If a patient says "I went for a long boat ride" and the psychiatrist says "Tell me about boats", you don't assume she didn't know what

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`

Figure 15.1 Sample ELIZA dialogue from Weizenbaum (1966).

a boat is, but rather assume she had some conversational goal.[1]

Weizenbaum made use of this property of Rogerian psychiatric conversations, along with clever regular expressions, to allow ELIZA to interact in ways that seemed deceptively human-like, as in the sample conversational fragment in Fig. 15.1.

As we foreshadowed in Chapter 2, ELIZA worked by simple rules roughly like:

```
(.*) YOU (.*) ME   ->  WHAT MAKES YOU THINK I \2 YOU
```

to transform a user sentence like "`You hate me`" into a system response like

```
WHAT MAKES YOU THINK I HATE YOU
```

Among Weizenbaum's clever tricks are the linking of each ELIZA pattern/rule to a keyword. Consider the following user sentence:

```
    I know everybody laughed at me
```

Because it has the word "I", this sentence could match the following rule whose keyword is *I*:

```
 I (.*)  ->  You say you \1
```

producing:

```
YOU SAY YOU KNOW EVERYBODY LAUGHED AT YOU
```

Weizenbaum points out, however, that a more powerful response would rely on the keyword "everybody", since someone using universals like *everybody* or *always* is probably thinking about a specific person or situation. So the ELIZA algorithm prefers to respond using patterns associated more specific keywords like *everybody*:

```
WHO IN PARTICULAR ARE YOU THINKING OF?
```

If no keyword matches, the algorithm chooses a non-committal response like "`PLEASE GO ON`", "`THAT'S VERY INTERESTING`", or "`I SEE`".

ELIZA illustrates a number of important issues with chatbots. First, people became deeply emotionally involved and conducted very personal conversations, even to the extent of asking Weizenbaum to leave the room while they were typing. Reeves and Nass (1996) show that people tend to assign human characteristics to computers and interact with them in ways that are typical of human-human interactions. They interpret an utterance in the way they would if it had spoken by a human, (even though they are aware they are talking to a computer). This means that chatbots can have significant influences on people's cognitive and emotional state.

A second related issue is privacy. When Weizenbaum suggested that he might want to store the ELIZA conversations, people immediately pointed out that this would violate people's privacy. Modern chatbots in the home are likely to overhear

[1] This is due to the Gricean principle of *relevance* that we'll discuss in the next section..

private information, even if they aren't used for counseling as ELIZA was. Indeed, 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 (Ischen et al., 2019).

Both of these issues (emotional engagement and privacy) mean we need to think carefully about how we deploy chatbots and the people who are interacting with them. Dialogue research that uses human participants often requires getting permission from the Institutional Review Board (IRB) of your institution.

In the next section we introduce some basic properties of human conversation. We then turn in the rest of the chapter to the two basic paradigms for conversational interaction: frame-based dialogue systems and chatbots.

15.1 Properties of Human Conversation

Conversation between humans is an intricate and complex joint activity. Before we attempt to design a dialogue system 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. 15.2.

C_1: ...I need to travel in May.

A_2: And, what day in May did you want to travel?

C_3: OK uh I need to be there for a meeting that's from the 12th to the 15th.

A_4: And you're flying into what city?

C_5: Seattle.

A_6: And what time would you like to leave Pittsburgh?

C_7: Uh hmm I don't think there's many options for non-stop.

A_8: Right. There's three non-stops today.

C_9: What are they?

A_{10}: 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.

C_{11}: OK I'll take the 5ish flight on the night before on the 11th.

A_{12}: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.

C_{13}: OK.

A_{14}: And you said returning on May 15th?

C_{15}: Uh, yeah, at the end of the day.

A_{16}: OK. There's #two non-stops ...#

C_{17}: #Act...actually #, what day of the week is the 15th?

A_{18}: It's a Friday.

C_{19}: Uh hmm. I would consider staying there an extra day til Sunday.

A_{20}: OK...OK. On Sunday I have ...

Figure 15.2 Part of a phone conversation between a human travel agent (A) and human client (C). The passages framed by # in A_{16} and C_{17} indicate overlaps in speech.

Turns

turn

A dialogue is a sequence of **turns** (C_1, A_2, C_3, and so on), each a single contribution from one speaker to the dialogue (as if in a game: I take a turn, then you take a turn, then me, and so on). There are 20 turns in Fig. 15.2. A turn can consist of a sentence (like C_1), although it might be as short as a single word (C_{13}) or as long as multiple sentences (A_{10}).

endpointing

Turn structure has important implications for spoken dialogue. A human has to know when to stop talking; the client interrupts (in A_{16} and C_{17}), so a system that was performing this role must know to stop talking (and that the user might be making a correction). A system also has to know when to start talking. For example, most of the time in conversation, speakers start their turns almost immediately after the other speaker finishes, without a long pause, because people are can usually predict when the other person is about to finish talking. Spoken dialogue systems must also detect whether a user is done speaking, so they can process the utterance and respond. This task—called **endpointing** or **endpoint detection**— can be quite challenging because of noise and because people often pause in the middle of turns.

Speech Acts

speech acts

A key insight into conversation—due originally to the philosopher Wittgenstein (1953) but worked out more fully by Austin (1962)—is that each utterance in a dialogue is a kind of **action** being performed by the speaker. These actions are commonly called **speech acts** or **dialogue acts**: here's one taxonomy consisting of 4 major classes (Bach and Harnish, 1979):

Constatives: committing the speaker to something's being the case (*answering*, *claiming*, *confirming*, *denying*, *disagreeing*, *stating*)

Directives: attempts by the speaker to get the addressee to do something (*advising*, *asking*, *forbidding*, *inviting*, *ordering*, *requesting*)

Commissives: committing the speaker to some future course of action (*promising, planning, vowing, betting, opposing*)

Acknowledgments: express the speaker's attitude regarding the hearer with respect to some social action (*apologizing*, *greeting*, *thanking*, *accepting an acknowledgment*)

A user asking a person or a dialogue system to do something ('Turn up the music') is issuing a DIRECTIVE. Asking a question that requires an answer is also a way of issuing a DIRECTIVE: in a sense when the system says (A_2) "what day in May did you want to travel?" it's as if the system is (very politely) commanding the user to answer. By contrast, a user stating a constraint (like C_1 'I need to travel in May') is issuing a CONSTATIVE. A user thanking the system is issuing an ACKNOWLEDGMENT. The speech act expresses an important component of the intention of the speaker (or writer) in saying what they said.

Grounding

common ground

grounding

A dialogue is not just a series of independent speech acts, but rather a collective act performed by the speaker and the hearer. Like all collective acts, it's important for the participants to establish what they both agree on, called the **common ground** (Stalnaker, 1978). Speakers do this by **grounding** each other's utterances. Grounding means acknowledging that the hearer has understood the speaker (Clark, 1996). (People need grounding for non-linguistic actions as well; the reason an elevator button lights up when it's pressed is to acknowledge that the elevator has indeed been called, essentially grounding your action of pushing the button (Norman, 1988).)

Humans constantly ground each other's utterances. We can ground by explicitly saying "OK", as the agent does in A_8 or A_{10}. Or we can ground by repeating what the other person says; in utterance A_2 the agent repeats "in May", demonstrating her understanding to the client. Or notice that when the client answers a question, the agent begins the next question with “And”. The “And” implies that the new question is ‘in addition’ to the old question, again indicating to the client that the agent has successfully understood the answer to the last question.

Subdialogues and Dialogue Structure

conversation analysis

adjacency pair

Conversations have structure. Consider, for example, the local structure between speech acts discussed in the field of **conversation analysis** (Sacks et al., 1974). QUESTIONS set up an expectation for an ANSWER. PROPOSALS are followed by ACCEPTANCE (or REJECTION). COMPLIMENTS (“Nice jacket!”) often give rise to DOWNPLAYERS (“Oh, this old thing?”). These pairs, called **adjacency pairs** are composed of a **first pair part** and a **second pair part** (Schegloff, 1968), and these expectations can help systems decide what actions to take.

side sequence

subdialogue

However, dialogue acts aren’t always followed immediately by their second pair part. The two parts can be separated by a **side sequence** (Jefferson 1972) or **subdialogue**. For example utterances C_{17} to A_{20} constitute a **correction subdialogue** (Litman 1985, Litman and Allen 1987, Chu-Carroll and Carberry 1998):

C_{17}: #Act. . . actually#, what day of the week is the 15th?
A_{18}: It’s a Friday.
C_{19}: Uh hmm. I would consider staying there an extra day til Sunday.
A_{20}: OK. . . OK. On Sunday I have . . .

The question in C_{17} interrupts the prior discourse, in which the agent was looking for a May 15 return flight. The agent must answer the question and also realize that ‘‘I would consider staying...til Sunday” means that the client would probably like to change their plan, and now go back to finding return flights, but for the 17th.

Another side sequence is the **clarification question**, which can form a subdialogue between a REQUEST and a RESPONSE. This is especially common in dialogue systems where speech recognition errors causes the system to have to ask for clarifications or repetitions like the following:

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...

presequence

In addition to side-sequences, questions often have **presequences**, like the following example where a user starts with a question about the system’s capabilities (“Can you make train reservations”) before making a request.

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.

Initiative

initiative

Sometimes a conversation is completely controlled by one participant. For example a reporter interviewing a chef might ask questions, and the chef responds. We say that the reporter in this case has the conversational **initiative** (Carbonell, 1970; Nickerson, 1976). In normal human-human dialogue, however, it’s more common for initiative to shift back and forth between the participants, as they sometimes answer questions, sometimes ask them, sometimes take the conversations in new directions, sometimes not. You may ask me a question, and then I respond asking you to clarify something you said, which leads the conversation in all sorts of ways. We call such interactions **mixed initiative** (Carbonell, 1970).

Full mixed initiative, while the norm for human-human conversations, can be difficult for dialogue systems. The most primitive dialogue systems tend to use **system-initiative**, where the system asks a question and the user can't do anything until they answer it, or *user-initiative* like simple search engines, where the user specifies a query and the system passively responds. Even modern large language model-based dialogue systems, which come much closer to using full mixed initiative, often don't have completely natural initiative switching. Getting this right is an important goal for modern systems.

Inference and Implicature

Inference is also important in dialogue understanding. Consider the client's response C_2, repeated here:

A_2: And, what day in May did you want to travel?

C_3: OK uh I need to be there for a meeting that's from the 12th to the 15th.

Notice that the client does not in fact answer the agent's question. The client merely mentions a meeting at a certain time. What is it that licenses the agent to infer that the client is mentioning this meeting so as to inform the agent of the travel dates?

The speaker seems to expect the hearer to draw certain inferences; in other words, the speaker is communicating more information than seems to be present in the uttered words. This kind of example was pointed out by Grice (1975, 1978) as part of his theory of **conversational implicature**. **Implicature** means a particular class of licensed inferences. Grice proposed that what enables hearers to draw these inferences is that conversation is guided by a set of **maxims**, general heuristics that play a guiding role in the interpretation of conversational utterances. One such maxim is the maxim of **relevance** which says that speakers attempt to be relevant, they don't just utter random speech acts. When the client mentions a meeting on the 12th, the agent reasons 'There must be some relevance for mentioning this meeting. What could it be?'. The agent knows that one precondition for having a meeting (at least before Web conferencing) is being at the place where the meeting is held, and therefore that maybe the meeting is a reason for the travel, and if so, then since people like to arrive the day before a meeting, the agent should infer that the flight should be on the 11th.

implicature

relevance

These subtle characteristics of human conversations (**turns**, **speech acts**, **grounding**, **dialogue structure**, **initiative**, and **implicature**) are among the reasons it is difficult to build dialogue systems that can carry on natural conversations with humans. Many of these challenges are active areas of dialogue systems research.

15.2 Frame-Based Dialogue Systems

A **task-based dialogue** system has the goal of helping a user solve a specific task like making a travel reservation or buying a product. Task-based dialogue systems are based around **frames**, first introduced in the early influential **GUS** system for travel planning (Bobrow et al., 1977). Frames are knowledge structures representing the details of the user's task specification. Each frame consists of a collection of **slots**, each of which can take a set of possible **values**. Together a set of frames is

frame

GUS

slot

sometimes called a **domain ontology**.

Here we'll describe the most well-studied frame-based architecture, the **dialogue-state** architecture, made up of the six components shown in Fig. 15.3. In the next sections we'll introduce four of them, after introducing the idea of frames (deferring the speech recognition and synthesis components to Chapter 16).

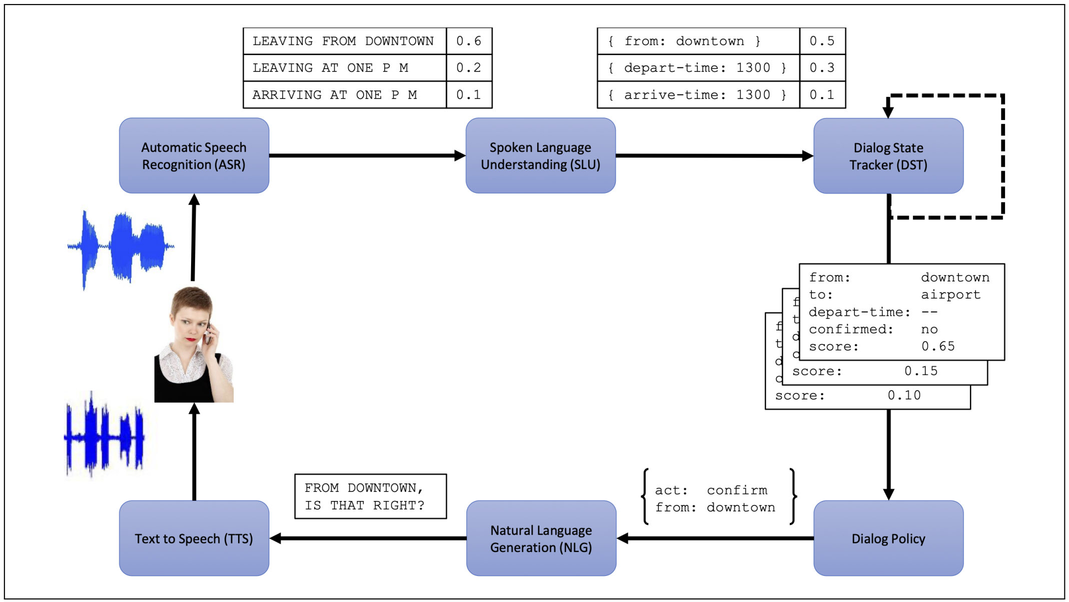

LEAVING FROM DOWNTOWN 0.6
LEAVING AT ONE P M 0.2
ARRIVING AT ONE P M 0.1
{ from: downtown } 0.5
{ depart-time: 1300 } 0.3
{ arrive-time: 1300 } 0.1
Automatic Speech Recognition (ASR)
Spoken Language Understanding (SLU)
Dialog State Tracker (DST)
from: downtown
to: airport
depart-time: --
confirmed: no
score: 0.65
score: 0.15
score: 0.10
act: confirm
from: downtown
FROM DOWNTOWN, IS THAT RIGHT?
Text to Speech (TTS)
Natural Language Generation (NLG)
Dialog Policy

Figure 15.3 Architecture of a dialogue-state system for task-oriented dialogue from Williams et al. (2016).

15.2.1 Frames and Slot Filling

The frame and its slots in a task-based dialogue system specify what the system needs to know to perform its task. A hotel reservation system needs dates and locations. An alarm clock system needs a time. The system's goal is to fill the slots in the frame with the fillers the user intends, and then perform the relevant action for the user (answering a question, or booking a flight).

Fig. 15.4 shows a sample frame for booking air travel, with some sample questions used for filling slots. In the simplest frame-based systems (including most commercial assistants until quite recently), these questions are pre-written templates, but in more sophisticated systems, questions are generated on-the-fly. The slot fillers are often constrained to a particular semantic type, like type CITY (taking on values like *San Francisco*, or *Hong Kong*) or DATE, AIRLINE, or TIME.

Slot	**Type**	Example Question
ORIGIN CITY	city	"From what city are you leaving?"
DESTINATION CITY	city	"Where are you going?"
DEPARTURE TIME	time	"When would you like to leave?"
DEPARTURE DATE	date	"What day would you like to leave?"
ARRIVAL TIME	time	"When do you want to arrive?"
ARRIVAL DATE	date	"What day would you like to arrive?"

Figure 15.4 A frame in a frame-based dialogue system, showing the type of each slot and a sample question used to fill the slot.

Many domains require multiple frames. Besides frames for car or hotel reservations, we might need other frames for things like general route information (for questions like *Which airlines fly from Boston to San Francisco?*), That means the system must be able to disambiguate which slot of which frame a given input is supposed to fill.

intent determination

slot filling

The task of slot-filling is usually combined with two other tasks, to extract 3 things from each user utterance. The first is **domain classification**: is this user for example talking about airlines, programming an alarm clock, or dealing with their calendar? The second is user **intent determination**: what general task or goal is the user trying to accomplish? For example the task could be to Find a Movie, or Show a Flight, or Remove a Calendar Appointment. Together, the domain classification and intent determination tasks decide which frame we are filling. Finally, we need to do **slot filling** itself: extract the particular slots and fillers that the user intends the system to understand from their utterance with respect to their intent. From a user utterance like this one:

```
Show me morning flights from Boston to San Francisco on Tuesday
```

a system might want to build a representation like:

DOMAIN:	AIR-TRAVEL	INTENT:	SHOW-FLIGHTS
ORIGIN-CITY:	Boston	DEST-CITY:	San Francisco
ORIGIN-DATE:	Tuesday	ORIGIN-TIME:	morning

Similarly an utterance like this:

```
Wake me tomorrow at 6
```

should give an intent like this:

DOMAIN:	ALARM-CLOCK
INTENT:	SET-ALARM
TIME:	2017-07-01 0600

The simplest dialogue systems use handwritten rules for slot-filling, like this regular expression for recognizing the SET-ALARM intent:

```
wake me (up) | set (the|an) alarm | get me up
```

But most systems use supervised machine-learning: each sentence in a training set is annotated with slots, domain, and intent, and a sequence model maps from input words to slot fillers, domain and intent. For example we'll have pairs of sentences that are labeled for domain (AIRLINE) and intent (SHOWFLIGHT), and are also labeled with BIO representations for the slots and fillers. (Recall from Chapter 17 that in 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.)

```
 O O    O  O    O  B-DES I-DES     O  B-DEPTIME I-DEPTIME  O        AIRLINE-SHOWFLIGHT
I want to fly to San   Francisco on Monday    afternoon  please   EOS
```

Fig. 15.5 shows a typical architecture for inference. The input words $w_1...w_n$ are passed through a pretrained language model encoder, followed by a feedforward layer and a softmax at each token position over possible BIO tags, with the output a series of BIO tags $s_1...s_n$. We generally combine the domain-classification and intent-extraction tasks with slot-filling 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

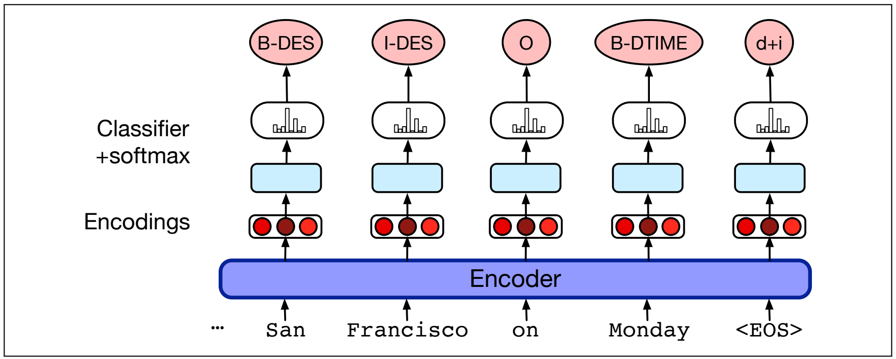


Figure 15.5 Slot filling by passing input words through an encoder, and then using a linear or feedforward layer followed by a softmax to generate a series of BIO tags. Here we also show a final state: a domain concatenated with an intent.

code 'SFO'), for example with dictionaries that specify that SF, SFO, and San Francisco are synonyms. Often in industrial contexts, combinations of rules and machine learning are used for each of these components.

We can make a very simple frame-based dialogue system by wrapping a small amount of code around this slot extractor. Mainly we just need to ask the user questions until all the slots are full, do a database query, then report back to the user, using hand-built templates for generating sentences.

15.2.2 Evaluating Task-Based Dialogue

task error rate

slot error rate

We evaluate task-based systems by computing the **task error rate**, or **task success rate**: the percentage of times the system booked the right plane flight, or put the right event on the calendar. A more fine-grained, but less extrinsic metric is the **slot error rate**, the percentage of slots filled with the correct values:

$$\text{Slot Error Rate for a Sentence} = \frac{\text{\# of inserted/deleted/subsituted slots}}{\text{\# of total reference slots for sentence}} \tag{15.1}$$

For example a system that extracted the slot structure below from this sentence:

(15.2) Make an appointment with Chris at 10:30 in Gates 104

Slot	Filler
PERSON	Chris
TIME	11:30 a.m.
ROOM	Gates 104

efficiency costs

has a slot error rate of 1/3, since the TIME is wrong. Instead of error rate, slot precision, recall, and F-score can also be used. We can also measure **efficiency costs** like the length of the dialogue in seconds or turns.

15.3 Dialogue Acts and Dialogue State

While the naive slot-extractor system described above can handle simple dialogues, often we want more complex interactions. For example, we might want to confirm that we've understand the user, or ask them to repeat themselves. We can build a more sophisticated system using **dialogue acts** and **dialogue state**.

15.3.1 Dialogue Acts

dialogue acts

Dialogue acts are a generalization of speech acts that also represent grounding. The set of acts can be general, or can be designed for particular dialogue tasks.

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

Figure 15.6 Dialogue acts used by the HIS restaurant recommendation system of Young et al. (2010). The Sys and User columns indicate which acts are valid as system outputs and user inputs, respectively.

Figure 15.6 shows a tagset for a restaurant recommendation system, and Fig. 15.7 shows these tags labeling a sample dialogue from the HIS system (Young et al., 2010). This example also shows the content of each dialogue act, which are the slot fillers being communicated. So the user might INFORM the system that they want Italian food near a museum, or CONFIRM with the system that the price is reasonable.

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

Figure 15.7 A dialogue from the HIS System of Young et al. (2010) using the dialogue acts in Fig. 15.6.

15.3.2 Dialogue State Tracking

The job of the dialogue-state tracker is to determine the current state of the frame (the fillers of each slot), and the user's most recent dialogue act. The dialogue-state is not just the slot-fillers in the current sentence; it includes the entire state of the frame at this point, summarizing all of the user's constraints. Fig. 15.8 from Mrkšić et al. (2017) shows the dialogue state after each turn.

Dialogue act detection is done just like domain or intent classification, by passing the input sentence through an encoder and adding an act classifier. Often passing in the prior dialogue act as well can improve classification. And since dialogue acts

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

Figure 15.8 The output of the dialogue state tracker after each turn (Mrkšić et al., 2017).

place some constraints on the slots and values, the tasks of dialogue-act detection and slot-filling are often performed jointly. The state tracker can just take the output of a slot-filling sequence-model (Section 15.2.1) after each sentence, or do something more complicated like training a classifier to decide if a value has been changed.

user correction acts

hyperarticulation

A special case: detecting correction acts. If a dialogue system misrecognizes or misunderstands an utterance, users will repeat or reformulate the utterance. Detecting these **user correction acts** is quite important, especially for spoken language. Ironically, corrections are actually *harder* to recognize than normal sentences (Swerts et al., 2000), because users who are frustrated adjust their speech in a way that is difficult for speech recognizers (Goldberg et al., 2003). For example speakers often use a prosodic style for corrections called **hyperarticulation**, in which the utterance is louder or longer or exaggerated in pitch, such as *I said BAL-TI-MORE, not Boston* (Wade et al. 1992, Levow 1998, Hirschberg et al. 2001). Detecting acts can be part of the general dialogue act detection classifier, or can make use of special features beyond the words, like those shown below (Levow 1998, Litman et al. 1999, Hirschberg et al. 2001, Bulyko et al. 2005, Awadallah et al. 2015).

features	**examples**
semantic	embedding similarity between correction and user's prior utterance
phonetic	phonetic overlap between candidate correction act and user's prior utterance (i.e. "WhatsApp" may be incorrectly recognized as "What's up")
prosodic	hyperarticulation, increases in F0 range, pause duration, and word duration
ASR	ASR confidence, language model probability

15.3.3 Dialogue Policy: Which act to generate

dialogue policy

content planning

In early commercial frame-based systems, the dialogue policy is simple: ask questions until all the slots are full, do a database query, then report back to the user. A more sophisticated **dialogue policy** can help a system decide when to answer the user's questions, when to instead ask the user a clarification question, and so on. A dialogue policy thus decides what dialogue act to generate. Choosing a dialogue act to generate, along with its arguments, is sometimes called **content planning**.

explicit confirmation

Let's see how to do this for some important dialogue acts. Dialogue systems, especially speech systems, often misrecognize the users' words or meaning. To ensure system and user share a common ground, systems must **confirm** understandings with the user or **reject** utterances that the system don't understand. A system might use an **explicit confirmation** act to confirm with the user, like **Is that correct?** below:

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?**

implicit confirmation

When using an **implicit confirmation** act, a system instead grounds more implicitly, for example by repeating the system’s understanding as part of asking the next question, as *Shanghai* is confirmed in passing in this example:

U: I want to travel to to Shanghai
S: **When do you want to travel to Shanghai?**

rejection

There’s a tradeoff. Explicit confirmation makes it easier for users to correct misrecognitions by just answering “no” to the confirmation question. But explicit confirmation is time-consuming and awkward (Danieli and Gerbino 1995, Walker et al. 1998). We also might want an act that expresses lack of understanding: **rejection**, for example with a prompt like *I’m sorry, I didn’t understand that*. To decide among these acts, we can make use of the fact that ASR systems often compute their **confidence** in their transcription (often based on the log-likelihood the system assigns the sentence). A system can thus choose to explicitly confirm only low-confidence sentences. Or systems might have a four-tiered level of confidence with three thresholds α, β, and γ:

$< \alpha$	low confidence	reject
$\geq \alpha$	above the threshold	confirm explicitly
$\geq \beta$	high confidence	confirm implictly
$\geq \gamma$	very high confidence	don’t confirm at all

15.3.4 Natural language generation: Sentence Realization

	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.

Figure 15.9 Sample inputs to the sentence realization phase of NLG, showing the dialogue act and attributes prespecified by the content planner, and two distinct potential output sentences to be generated. From the restaurant recommendation system of Nayak et al. (2017).

sentence realization

delexicalize

Once a dialogue act has been chosen, we need to generate the text of the response to the user. This part of the generation process is called **sentence realization**. Fig. 15.9 shows a sample input/output for the sentence realization phase. The content planner has chosen the dialogue act RECOMMEND and some slots (name, neighborhood, cuisine) and fillers. The sentence realizer generates a sentence like lines 1 or 2 (by training on examples of representation/sentence pairs from a corpus of labeled dialogues). Because we won’t see every restaurant or attribute in every possible wording, we can **delexicalize**: generalize the training examples by replacing specific slot value words in the training set with a generic placeholder token representing the slot. Fig. 15.10 shows the sentences in Fig. 15.9 delexicalized.

We can map from frames to delexicalized sentences with an encoder decoder model (Mrkšić et al. 2017, inter alia), trained on hand-labeled dialogue corpora like MultiWOZ (Budzianowski et al., 2018). The input to the encoder is a sequence of 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.

Figure 15.10 Delexicalized sentences that can be used for generating many different relexicalized sentences. From the restaurant recommendation system of Nayak et al. (2017).

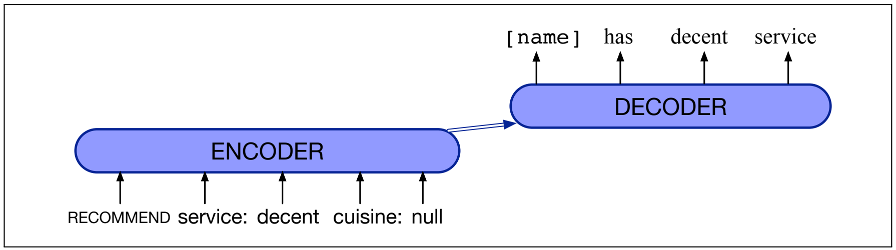


Figure 15.11 An encoder decoder sentence realizer mapping slots/fillers to English.

tokens x_t that represent the dialogue act (e.g., RECOMMEND) and its arguments (e.g., service:decent, cuisine:null) (Nayak et al., 2017), as in Fig. 15.11.

relexicalize

The decoder outputs the delexicalized English sentence "name has decent service", which we can then **relexicalize**, i.e. fill back in correct slot values, resulting in "Au Midi has decent service".

15.4 Chatbots

chatbot

Chatbots are systems that can carry on extended conversations with the goal of mimicking the unstructured conversations or 'chats' characteristic of informal human-human interaction. While early systems like ELIZA (Weizenbaum, 1966) or PARRY (Colby et al., 1971) had theoretical goals like testing theories of psychological counseling, for most of the last 50 years chatbots have been designed for entertainment. That changed with the recent rise of neural chatbots like ChatGPT, which incorporate solutions to NLP tasks like question answering, writing tools, or machine translation into a conversational interface. A conversation with ChatGPT is shown in Fig. 15.12. In this section we describe neural chatbot architectures and datasets.

[TBD]

Figure 15.12 A conversation with ChatGPT.

15.4.1 Training chatbots

Data Chatbots are generally trained on a training set that includes standard large language model training data of the type discussed in Section **??**: versions of the web from the Common Crawl, including news sites, Wikipedia, as well as books. For training chatbots, it is common to additionally add lots of dialogue data.

This can include datasets created specifically for training chatbots by hiring speakers of the language to have conversations, such as by having them take on personas or talk about knowledge provided to them. For example the Topical-Chat dataset has 11K crowdsourced conversations spanning 8 broad topics (Gopalakrishnan et al., 2019), the EMPATHETICDIALOGUES includes 25K crowdsourced conversations grounded in a specific situation where a speaker was feeling a specific emotion (Rashkin et al., 2019), and the SaFeRDialogues dataset (Ung et al., 2022) has 8k dialogues demonstrating graceful responses to conversational feedback about safety failures.

Such datasets are far too small to train a language model alone, and so it's common to also pretrain on large datasets of pseudo-conversations drawn from Twitter (Ritter et al., 2010), Reddit (Roller et al., 2021), Weibo (微博), and other social media platforms. To turn social media data into data that has the structure of a conversation, we can treat any post on the platform as the first turn in a conversation, and the sequence of comments/replies as subsequent turns in that conversation.

Datasets from the web can be enormously toxic, so it's crucial to filter the dialogues first. This can be done by using the same toxicity classifiers we describe below in the fine-tuning section.

Architecture For training chatbots, it's most common to use the standard causal language model architecture, in which the model predicts each word given all the prior words, and the loss is the standard language modeling loss. Fig. 15.13 shows a standard training setup; no different than language model training in Chapter 9. The only difference is the data, which has the addition of significant conversation and pseudo-conversation data as described in the prior section. As usual, the left context can include the entire prior conversation (or as much as fits in the context window).

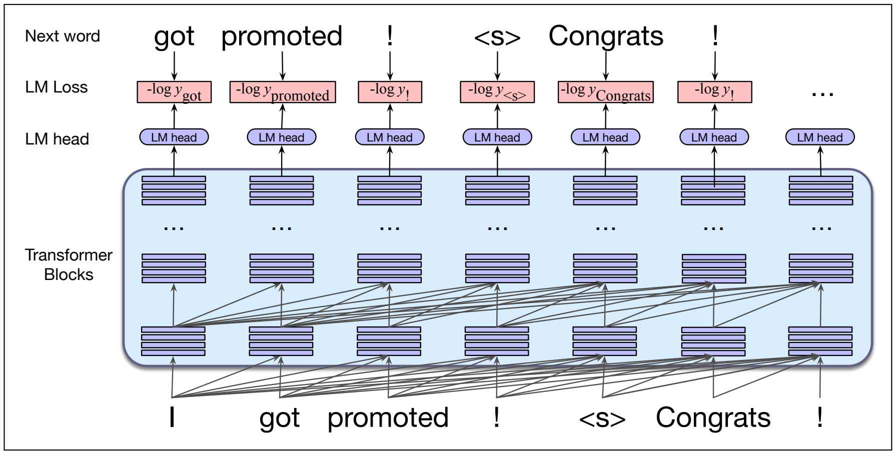


Figure 15.13 Training a causal (decoder-only) language model for a chatbot.

An alternative is to use the encoder-decoder architecture of Chapter 13. In this case the entire conversation up to the last turn (as much as fits in the context) is presented to the encoder, and the decoder generates the next turn.

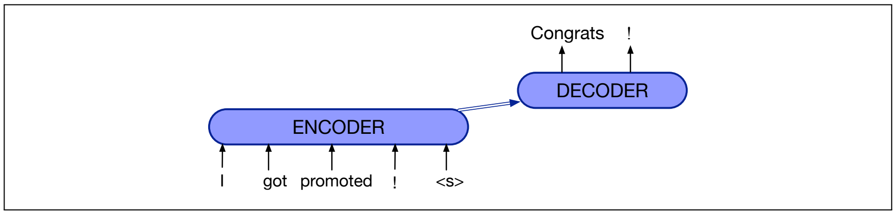


Figure 15.14 An alternative: an encoder-decoder language model for a chatbot.

In practice, dialogue systems require additional customization beyond just pretraining on dialogue data. In the next few sections we'll discuss various stages of fine-tuning that can be used for this customization.

15.4.2 Fine Tuning for Quality and Safety

It is a common practice for dialogue systems to use further labeled data for fine-tuning. One function of this fine-tuning step is to improve the quality of the dialogue, training the system to produce responses that are sensible and interesting. Another function might be to improve safety, keeping a dialogue system from suggesting harmful actions (like financial fraud, medical harm, inciting hatred, or abusing the user or other people).

In the simplest method for improving quality and safety, speakers of the language are given an initial prompt and instructions to have high-quality, safe dialogues. They then interact with an initial dialogue system and their responses are used to finetune the model, usually as part of the instruct tuning step we introduced in Chapter 12. Thus a dialogue system learns to answer questions, follow other instructions, and also carry on high-quality, safe dialogues, in a single multi-task learning format.

While fine-tuning on positive examples is helpful, it is generally insufficient and so it is common to add more discriminative data that specifically downweights low-quality or harmful responses. The simplest paradigm for this is to train a model to predict turn-level safety and quality values, by training on human-labeled ratings. Such ratings might be collected by first having speakers of the language carry on dialogues with a system, and then a second set of people act as labelers to label every system turn for its quality and safety, resulting in a binary label for quality and safety for each turn.

Once a dataset has been created with these labels, a language model can be used in a classification task to label the quality and safety of a turn. For example in the Lamda system (Cohen et al., 2022), a single language model is used in two phases, roughly corresponding to generative and discriminative tasks: first generating a response, and then generating a label. In the generative phase, the model is given the prior turn and a special RESPONSE token and generates the blue response turn. (In training, the training loss is given only for the blue response):

“What’s up? RESPONSE Not much.

In a second, discriminative phase, the model is fine-tuned to see an attribute (SENSIBLE, INTERESTING, UNSAFE) and then to predict a 0 or 1 value, again with training losses given only for the blue value.

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

To use the system in inference, the model first generates a response given the context, and then it is given the attribute and asked to generate a rating. The result is a generated turn along with a label. This label isn’t shown to the user but can be use for filtering, either at training time or at deployment time. For example, the system can generate multiple potential responses, filter out any response that is unsafe, and return to the user the highest ranking response.

15.4.3 Learning to perform retrieval as part of responding

Modern chatbots are also trained to do retrieval (calls to Google or Bing or other search engines) as a component of generating a response. This is generally done by having a sort of fake dialogue participant called something like `Search Query` whose utterances are passed to a search engine. The system can be encourged to do searches either by prompting or by fine-tuning.

For example in the Sparrow chatbot (Glaese et al., 2022), the prompt includes a few sample conversations that include two special users, `Search Query`, and `Search Results`:

	...[this is preceded by other parts of the prompt]
User:	What color is the sky?
Sparrow:	Blue.
User:	Why?
Search Query:	why is sky blue
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.
Sparrow:	Because of the Rayleigh scattering. It causes short waves of blue light to be scattered more than the other colours in the spectrum.
User:	Who was Raleigh?
Search Query:	who was Rayleigh
...	...

From these prompts, the system learns to generate texts with `Search Query` turns for fact-based questions from the user, and these are passed to a search engine to generate the `Search Results` turns.

Alternatively, systems can be finetuned to to know when to use a search engine. For example, labelers can interact with a system, fact check each of the responses, and whenever the system emits an incorrect response, perform the web search queries that the system should have used to check its answer, and then the interation is recorded and used for fine-tuning. Or labelers can look at a transcript of a language model carrying on a dialogue, and similarly mark every place where a fact was wrong (or out-of-date) and write the set of search queries that would have been appropriate. A system is then fine-tuned to generate `search query` turns which are again passed to a search engine to generate the `search responses`. The set of pages or snippets returned by the search engine in the search response turn are then treated as the context for generation, similarly to the retrieval-based question-answering methods of Chapter 14.

15.4.4 RLHF

RLHF

A more sophisticated family of methods uses reinforcement learning to learn to match human preferences for generated turns. In this method, **RLHF** for Reinforcement Learning from Human Feedback, we give a system a dialogue context and sample two possible turns from the language model. We then have humans label which of the two is better, creating a large dataset of sentence pairs with human preferences. These pairs are used to train a dialogue policy, and reinforcement learning is used to train the language model to generate turns that have higher rewards (Christiano et al., 2017; Ouyang et al., 2022). While using RLHF is the current state of the art at the time of this writing, a number of alternatives have been recently developed that don't require reinforcement learning (Rafailov et al., 2023, e.g.,) and so this aspect of the field is changing very quickly.

15.4.5 Evaluating Chatbots

Chatbots are evaluated by humans, who assign a score. This can be the human who talked to the chatbot (**participant evaluation**) or a third party who reads a transcript of a human/chatbot conversation (**observer evaluation**). In the participant evaluation of See et al. (2019), the human evaluator chats with the model for six turns and rates the chatbot on 8 dimensions capturing conversational quality: avoiding repetition, interestingness, making sense, fluency, listening, inquisitiveness, humanness and engagingness on Likert scales like these:

Engagingness How much did you enjoy talking to this user?
• Not at all • A little • Somewhat • A lot

Making sense How often did this user say something which did NOT make sense?
• Never made any sense • Most responses didn't make sense • Some responses didn't make sense • Everything made perfect sense

Observer evaluations use third party annotators to look at the text of a complete conversation. Sometimes we're interested in having raters assign a score to each system turn; for example (Artstein et al., 2009) have raters mark how *coherent* each turn is. Often, however, we just want a single high-level score to know if system A is better than system B. The **acute-eval** metric (Li et al., 2019) is such an observer evaluation in which annotators look at two separate human-computer conversations and choose the system which performed better on four metrics: engagingness, interestingness, humanness, and knowledgability.

acute-eval

15.5 Dialogue System Design

Because of the important role of the user, the field of dialogue systems is closely linked with Human-Computer Interaction (HCI). This is especially true for task-oriented dialogue and assistants, where the design of dialogue strategies, sometimes called **voice user interface** design, generally follows **user-centered design** principles (Gould and Lewis, 1985):

voice user interface

1. Study the user and task: Understand the users and the task by interviewing users, investigating similar systems, and studying related human-human dialogues.

2. Build simulations and prototypes: A crucial tool in building dialogue systems is the **Wizard-of-Oz system**. In wizard systems, the users interact with what they

Wizard-of-Oz system

think is a program but is in fact a human "wizard" disguised by a software interface (Gould et al. 1983, Good et al. 1984, Fraser and Gilbert 1991). The name comes from the children's book *The Wizard of Oz* (Baum, 1900), in which the wizard turned out to be a simulation controlled by a man behind a curtain or screen. A wizard system can be used to test out an architecture before implementation; only the interface software and databases need to be in place. The wizard gets input from the user, uses a database interface to run queries based on the user utterance, and then outputs sentences, either by typing them or speaking them.

Wizard-of-Oz systems are not a perfect simulation, since the wizard doesn't exactly simulate the errors or limitations of a real system; but wizard studies can still provide a useful first idea of the domain issues.

barged in

value sensitive design

3. Iteratively test the design on users: An iterative design cycle with embedded user testing is essential in system design (Nielsen 1992, Cole et al. 1997, Yankelovich et al. 1995, Landauer 1995). For example in a well-known incident, an early dialogue system required the user to press a key to interrupt the system (Stifelman et al., 1993). But user testing showed users **barged in** (interrupted, talking over the system), which led to a redesign of the system to recognize overlapped speech. It's also important to incorporate **value sensitive design**, in which we carefully consider during the design process the benefits, harms and possible stakeholders of the resulting system (Friedman et al. 2017, Friedman and Hendry 2019).

15.5.1 Ethical Issues in Dialogue System Design

Ethical issues have been key to how we think about designing artificial agents since well before we had dialogue systems. Mary Shelley (depicted below) centered her novel *Frankenstein* around the problem of creating artificial agents without considering ethical and humanistic concerns. One issue is the **safety** of users. If users seek information from dialogue systems in safety-critical situations like asking medical advice, or in emergency situations, or when indicating the intentions of self-harm, incorrect advice can be dangerous and even life-threatening. For example (Bickmore et al., 2018) gave participants medical problems to pose to three commercial dialogue systems (Siri, Alexa, Google Assistant) and asked them to determine an action to take based on the system responses; many of the proposed actions, if actually taken, would have led to harm or death.

Tay

A system can also harm users by verbally attacking them, or creating *representational harms* (Blodgett et al., 2020) by generating abusive or harmful stereotypes that demean particular groups of people. Both abuse and stereotypes can cause psychological harm to users. Microsoft's 2016 **Tay** chatbot, for example, was taken offline 16 hours after it went live, when it began posting messages with racial slurs, conspiracy theories, and personal attacks on its users. Tay had learned these biases and actions from its training data, including from users who seemed to be purposely teaching the system to repeat this kind of language (Neff and Nagy 2016). Henderson et al. (2017) examined dialogue datasets used to train corpus-based chatbots and found toxic and abusive language, especially in social media corpora like Twitter and Reddit, and indeed such language then appears in the text generated by language models and dialogue systems (Gehman et al. 2020; Xu et al. 2020) which can even amplify the bias from the training data (Dinan et al., 2020). Liu et al. (2020) developed another method for investigating bias, testing how neural dialogue systems responded to pairs of simulated user turns that are identical except for mentioning different genders or race. They found, for example, that simple changes like using the word 'she' instead of 'he' in a sentence caused systems to respond more offensively and with more negative sentiment.

Another important ethical issue is **privacy**. Already in the first days of ELIZA, Weizenbaum pointed out the privacy implications of people's revelations to the chatbot. The ubiquity of in-home dialogue systems means they may often overhear private information (Henderson et al., 2017). If a chatbot is human-like, users are also more likely to disclose private information, and less likely to worry about the harm of this disclosure (Ischen et al., 2019). In general, chatbots that are trained on transcripts of human-human or human-machine conversation must anonymize personally identifiable information.

Finally, chatbots raise important issues of gender equality in addition to textual bias. Current chatbots are overwhelmingly given female names, likely perpetuating the stereotype of a subservient female servant (Paolino, 2017). And when users use sexually harassing language, most commercial chatbots evade or give positive responses rather than responding in clear negative ways (Fessler, 2017).

These ethical issues are an important area of investigation, including finding ways to mitigate problems of abuse and toxicity, like detecting and responding appropriately to toxic contexts (Wolf et al. 2017, Dinan et al. 2020, Xu et al. 2020). Value sensitive design, carefully considering possible harms in advance (Friedman et al. 2017, Friedman and Hendry 2019) is also important; (Dinan et al., 2021) give a number of suggestions for best practices in dialogue system design. For example getting informed consent from participants, whether they are used for training, or whether they are interacting with a deployed system is important. Because dialogue systems by definition involve human participants, researchers also work on these issues with the Institutional Review Boards (**IRB**) at their institutions, who help protect the safety of experimental subjects.

IRB

15.6 Summary

Chatbots and **dialogue systems** are crucial speech and language processing applications that are already widely used commercially.

- In human dialogue, speaking is a kind of action; these acts are referred to as speech acts or dialogue acts. Speakers also attempt to achieve **common ground** by acknowledging that they have understand each other. Conversation also is characterized by turn structure and dialogue structure.
- Chatbots are conversational systems designed to mimic the appearance of informal human conversation. Rule-based chatbots like ELIZA and its modern

descendants use rules to map user sentences into system responses. Corpus-based chatbots mine logs of human conversation to learn to automatically map user sentences into system responses.

- For task-based dialogue, most commercial dialogue systems use the GUS or frame-based architecture, in which the designer specifies frames consisting of slots that the system must fill by asking the user.
- The **dialogue-state** architecture augments the GUS frame-and-slot architecture with richer representations and more sophisticated algorithms for keeping track of user's dialogue acts, **policies** for generating its own dialogue acts, and a natural language component.
- Dialogue systems are a kind of human-computer interaction, and general HCI principles apply in their design, including the role of the user, simulations such as Wizard-of-Oz systems, and the importance of iterative design and testing on real users.

Bibliographical and Historical Notes

The linguistic, philosophical, and psychological literature on dialogue is quite extensive. For example the idea that utterances in a conversation are a kind of **action** being performed by the speaker was due originally to the philosopher Wittgenstein (1953) but worked out more fully by Austin (1962) and his student John Searle. Various sets of speech acts have been defined over the years, and a rich linguistic and philosophical literature developed, especially focused on explaining the use of indirect speech acts. The idea of dialogue acts draws also from a number of other sources, including the ideas of adjacency pairs, pre-sequences, and other aspects of the interactional properties of human conversation developed in the field of **conversation analysis** (see Levinson (1983) for an introduction to the field). This idea that acts set up strong local dialogue expectations was also prefigured by Firth (1935, p. 70), in a famous quotation:

conversation analysis

> *Most of the give-and-take of conversation in our everyday life is stereotyped and very narrowly conditioned by our particular type of culture. It is a sort of roughly prescribed social ritual, in which you generally say what the other fellow expects you, one way or the other, to say.*

Another important research thread modeled dialogue as a kind of collaborative behavior, including the ideas of common ground (Clark and Marshall, 1981), reference as a collaborative process (Clark and Wilkes-Gibbs, 1986), joint intention (Levesque et al., 1990), and shared plans (Grosz and Sidner, 1980).

The earliest conversational systems were simple pattern-action chatbots like ELIZA (Weizenbaum, 1966). ELIZA had a widespread influence on popular perceptions of artificial intelligence, and brought up some of the first ethical questions in natural language processing —such as the issues of privacy we discussed above as well the role of algorithms in decision-making— leading its creator Joseph Weizenbaum to fight for social responsibility in AI and computer science in general.

Computational-implemented theories of dialogue blossomed in the 1970. That period saw the very influential GUS system (Bobrow et al., 1977), which in the late 1970s established the frame-based paradigm that became the dominant industrial paradigm for dialogue systems for over 30 years.

Another influential line of research from that decade focused on modeling the hierarchical structure of dialogue. Grosz's pioneering 1977 dissertation first showed that "task-oriented dialogues have a structure that closely parallels the structure of the task being performed" (p. 27), leading to her work with Sidner and others showing how to use similar notions of intention and plans to model discourse structure and coherence in dialogue. See, e.g., Lochbaum et al. (2000) for a summary of the role of intentional structure in dialogue.

Yet a third line, first suggested by Bruce (1975), suggested that since speech acts are actions, they should be planned like other actions, and drew on the AI planning literature (Fikes and Nilsson, 1971). A system seeking to find out some information can come up with the plan of asking the interlocutor for the information. A system hearing an utterance can interpret a speech act by running the planner "in reverse", using inference rules to infer from what the interlocutor said what the plan might have been. Plan-based models of dialogue are referred to as **BDI** models because such planners model the **beliefs**, **desires**, and **intentions** (BDI) of the system and interlocutor. BDI models of dialogue were first introduced by Allen, Cohen, Perrault, and their colleagues in a number of influential papers showing how speech acts could be generated (Cohen and Perrault, 1979) and interpreted (Perrault and Allen 1980, Allen and Perrault 1980). At the same time, Wilensky (1983) introduced plan-based models of understanding as part of the task of interpreting stories.

BDI

In the 1990s, machine learning models that had first been applied to natural language processing began to be applied to dialogue tasks like slot filling (Miller et al. 1994, Pieraccini et al. 1991). This period also saw lots of analytic work on the linguistic properties of dialogue acts and on machine-learning-based methods for their detection. (Sag and Liberman 1975, Hinkelman and Allen 1989, Nagata and Morimoto 1994, Goodwin 1996, Chu-Carroll 1998, Shriberg et al. 1998, Stolcke et al. 2000, Gravano et al. 2012. This work strongly informed the development of the dialogue-state model (Larsson and Traum, 2000). Dialogue state tracking quickly became an important problem for task-oriented dialogue, and there has been an influential annual evaluation of state-tracking algorithms (Williams et al., 2016).

The turn of the century saw a line of work on applying reinforcement learning to dialogue, which first came out of AT&T and Bell Laboratories with work on MDP dialogue systems (Walker 2000, Levin et al. 2000, Singh et al. 2002) along with work on cue phrases, prosody, and rejection and confirmation. Reinforcement learning research turned quickly to the more sophisticated POMDP models (Roy et al. 2000, Lemon et al. 2006, Williams and Young 2007) applied to small slot-filling dialogue tasks. Neural reinforcement learning models have been used both for chatbot systems, for example simulating dialogues between two dialogue systems, rewarding good conversational properties like coherence and ease of answering (Li et al., 2016), and for task-oriented dialogue (Williams et al., 2017).

By around 2010 the GUS architecture finally began to be widely used commercially in dialogue systems on phones like Apple's SIRI (Bellegarda, 2013) and other digital assistants.

The rise of the web gave rise to corpus-based chatbot architectures around the turn of the century, first using information retrieval models and then in the 2010s, after the rise of deep learning, with sequence-to-sequence models.

[TBD: Modern history of neural chatbots]

Other important dialogue areas include the study of affect in dialogue (Rashkin et al. 2019, Lin et al. 2019) and conversational interface design (Cohen et al. 2004, Harris 2005, Pearl 2017, Deibel and Evanhoe 2021).

Exercises

15.1 Write a finite-state automaton for a dialogue manager for checking your bank balance and withdrawing money at an automated teller machine.

dispreferred response

15.2 A **dispreferred response** is a response that has the potential to make a person uncomfortable or embarrassed in the conversational context; the most common example dispreferred responses is turning down a request. People signal their discomfort with having to say no with surface cues (like the word *well*), or via significant silence. Try to notice the next time you or someone else utters a dispreferred response, and write down the utterance. What are some other cues in the response that a system might use to detect a dispreferred response? Consider non-verbal cues like eye gaze and body gestures.

15.3 When asked a question to which they aren't sure they know the answer, people display their lack of confidence by cues that resemble other dispreferred responses. Try to notice some unsure answers to questions. What are some of the cues? If you have trouble doing this, read Smith and Clark (1993) and listen specifically for the cues they mention.

15.4 Implement a small air-travel help system based on text input. Your system should get constraints from users about a particular flight that they want to take, expressed in natural language, and display possible flights on a screen. Make simplifying assumptions. You may build in a simple flight database or you may use a flight information system on the Web as your backend.

Allen, J. and C. R. Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

Artstein, R., S. Gandhe, J. Gerten, A. Leuski, and D. Traum. 2009. Semi-formal evaluation of conversational characters. In *Languages: From Formal to Natural*, 22–35. Springer.

Austin, J. L. 1962. *How to Do Things with Words*. Harvard University Press.

Awadallah, A. H., R. G. Kulkarni, U. Ozertem, and R. Jones. 2015. Charaterizing and predicting voice query reformulation. *CIKM-15*.

Bach, K. and R. Harnish. 1979. *Linguistic communication and speech acts*. MIT Press.

Baum, L. F. 1900. *The Wizard of Oz*. Available at Project Gutenberg.

Bellegarda, J. R. 2013. Natural language technology in mobile devices: Two grounding frameworks. In *Mobile Speech and Advanced Natural Language Solutions*, 185–196. Springer.

Bickmore, T. W., H. Trinh, S. Olafsson, T. K. O'Leary, R. Asadi, N. M. Rickles, and R. Cruz. 2018. Patient and consumer safety risks when using conversational assistants for medical information: An observational study of Siri, Alexa, and Google Assistant. *Journal of Medical Internet Research*, 20(9):e11510.

Blodgett, S. L., S. Barocas, H. Daumé III, and H. Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. *ACL*.

Bobrow, D. G., R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd. 1977. GUS, A frame driven dialog system. *Artificial Intelligence*, 8:155–173.

Bruce, B. C. 1975. Generation as a social action. *Proceedings of TINLAP-1 (Theoretical Issues in Natural Language Processing)*.

Budzianowski, P., T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić. 2018. MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. *EMNLP*.

Bulyko, I., K. Kirchhoff, M. Ostendorf, and J. Goldberg. 2005. Error-sensitive response generation in a spoken language dialogue system. *Speech Communication*, 45(3):271–288.

Carbonell, J. R. 1970. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202.

Christiano, P. F., J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. 2017. Deep reinforcement learning from human preferences. *NeurIPS*, volume 30.

Chu-Carroll, J. 1998. A statistical model for discourse act recognition in dialogue interactions. *Applying Machine Learning to Discourse Processing. Papers from the 1998 AAAI Spring Symposium.* Tech. rep. SS-98-01. AAAI Press.

Chu-Carroll, J. and S. Carberry. 1998. Collaborative response generation in planning dialogues. *Computational Linguistics*, 24(3):355–400.

Clark, H. H. 1996. *Using Language*. Cambridge University Press.

Clark, H. H. and C. Marshall. 1981. Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, and I. A. Sag, eds, *Elements of Discourse Understanding*, 10–63. Cambridge.

Clark, H. H. and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

Cohen, A. D., A. Roberts, A. Molina, A. Butryna, A. Jin, A. Kulshreshtha, B. Hutchinson, B. Zevenbergen, B. H. Aguera-Arcas, C. ching Chang, C. Cui, C. Du, D. D. F. Adiwardana, D. Chen, D. D. Lepikhin, E. H. Chi, E. Hoffman-John, H.-T. Cheng, H. Lee, I. Krivokon, J. Qin, J. Hall, J. Fenton, J. Soraker, K. Meier-Hellstern, K. Olson, L. M. Aroyo, M. P. Bosma, M. J. Pickett, M. A. Menegali, M. Croak, M. Díaz, M. Lamm, M. Krikun, M. R. Morris, N. Shazeer, Q. V. Le, R. Bernstein, R. Rajakumar, R. Kurzweil, R. Thoppilan, S. Zheng, T. Bos, T. Duke, T. Doshi, V. Y. Zhao, V. Prabhakaran, W. Rusch, Y. Li, Y. Huang, Y. Zhou, Y. Xu, and Z. Chen. 2022. Lamda: Language models for dialog applications. ArXiv preprint.

Cohen, M. H., J. P. Giangola, and J. Balogh. 2004. *Voice User Interface Design*. Addison-Wesley.

Cohen, P. R. and C. R. Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212.

Colby, K. M., S. Weber, and F. D. Hilf. 1971. Artificial paranoia. *Artificial Intelligence*, 2(1):1–25.

Cole, R. A., D. G. Novick, P. J. E. Vermeulen, S. Sutton, M. Fanty, L. F. A. Wessels, J. H. de Villiers, J. Schalkwyk, B. Hansen, and D. Burnett. 1997. Experiments with a spoken dialogue system for taking the US census. *Speech Communication*, 23:243–260.

Danieli, M. and E. Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. *AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*.

Deibel, D. and R. Evanhoe. 2021. *Conversations with Things: UX Design for Chat and Voice*. Rosenfeld.

Dinan, E., G. Abercrombie, A. S. Bergman, S. Spruit, D. Hovy, Y.-L. Boureau, and V. Rieser. 2021. Anticipating safety issues in e2e conversational ai: Framework and tooling. ArXiv.

Dinan, E., A. Fan, A. Williams, J. Urbanek, D. Kiela, and J. Weston. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. *EMNLP*.

Fessler, L. 2017. We tested bots like Siri and Alexa to see who would stand up to sexual harassment. *Quartz*. Feb 22, 2017. `https://qz.com/911681/`.

Fikes, R. E. and N. J. Nilsson. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

Firth, J. R. 1935. The technique of semantics. *Transactions of the philological society*, 34(1):36–73.

Fraser, N. M. and G. N. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

Friedman, B. and D. G. Hendry. 2019. *Value Sensitive Design: Shaping Technology with Moral Imagination*. MIT Press.

Friedman, B., D. G. Hendry, and A. Borning. 2017. A survey of value sensitive design methods. *Foundations and Trends in Human-Computer Interaction*, 11(2):63–125.

Gehman, S., S. Gururangan, M. Sap, Y. Choi, and N. A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. *Findings of EMNLP*.

Glaese, A., N. McAleese, M. Trebacz, J. Aslanides, V. Firoiu, T. Ewalds, M. Rauh, L. Weidinger, M. Chadwick, P. Thacker, L. Campbell-Gillingham, J. Uesato, P.-S. Huang, R. Comanescu, F. Yang, A. See, S. Dathathri, R. Greig, C. Chen, D. Fritz, J. Sanchez Elias, R. Green, S. Mokrá, N. Fernando, B. Wu, R. Foley, S. Young, I. Gabriel, W. Isaac, J. Mellor, D. Hassabis, K. Kavukcuoglu, L. A. Hendricks, and G. Irving. 2022. Improving alignment of dialogue agents via targeted human judgements. ArXiv preprint.

Goldberg, J., M. Ostendorf, and K. Kirchhoff. 2003. The impact of response wording in error correction subdialogs. *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*.

Good, M. D., J. A. Whiteside, D. R. Wixon, and S. J. Jones. 1984. Building a user-derived interface. *CACM*, 27(10):1032–1043.

Goodwin, C. 1996. Transparent vision. In E. Ochs, E. A. Schegloff, and S. A. Thompson, eds, *Interaction and Grammar*, 370–404. Cambridge University Press.

Gopalakrishnan, K., B. Hedayatnia, Q. Chen, A. Gottardi, S. Kwatra, A. Venkatesh, R. Gabriel, and D. Hakkani-Tür. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. *INTERSPEECH*.

Gould, J. D., J. Conti, and T. Hovanyecz. 1983. Composing letters with a simulated listening typewriter. *CACM*, 26(4):295–308.

Gould, J. D. and C. Lewis. 1985. Designing for usability: Key principles and what designers think. *CACM*, 28(3):300–311.

Gravano, A., J. Hirschberg, and Š. Beňuš. 2012. Affirmative cue words in task-oriented dialogue. *Computational Linguistics*, 38(1):1–39.

Grice, H. P. 1975. Logic and conversation. In P. Cole and J. L. Morgan, eds, *Speech Acts: Syntax and Semantics Volume 3*, 41–58. Academic Press.

Grice, H. P. 1978. Further notes on logic and conversation. In P. Cole, ed., *Pragmatics: Syntax and Semantics Volume 9*, 113–127. Academic Press.

Grosz, B. J. 1977. *The Representation and Use of Focus in Dialogue Understanding*. Ph.D. thesis, University of California, Berkeley.

Grosz, B. J. and C. L. Sidner. 1980. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, eds, *Intentions in Communication*, 417–444. MIT Press.

Guindon, R. 1988. A multidisciplinary perspective on dialogue structure in user-advisor dialogues. In R. Guindon, ed., *Cognitive Science and Its Applications for Human-Computer Interaction*, 163–200. Lawrence Erlbaum.

Harris, R. A. 2005. *Voice Interaction Design: Crafting the New Conversational Speech Systems*. Morgan Kaufmann.

Henderson, P., K. Sinha, N. Angelard-Gontier, N. R. Ke, G. Fried, R. Lowe, and J. Pineau. 2017. Ethical challenges in data-driven dialogue systems. *AAAI/ACM AI Ethics and Society Conference*.

Hinkelman, E. A. and J. Allen. 1989. Two constraints on speech act ambiguity. *ACL*.

Hirschberg, J., D. J. Litman, and M. Swerts. 2001. Identifying user corrections automatically in spoken dialogue systems. *NAACL*.

Ischen, C., T. Araujo, H. Voorveld, G. van Noort, and E. Smit. 2019. Privacy concerns in chatbot interactions. *International Workshop on Chatbot Research and Design*.

Jefferson, G. 1972. Side sequences. In D. Sudnow, ed., *Studies in social interaction*, 294–333. Free Press, New York.

Landauer, T. K., ed. 1995. *The Trouble with Computers: Usefulness, Usability, and Productivity*. MIT Press.

Larsson, S. and D. R. Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, 6(323-340):97–114.

Lemon, O., K. Georgila, J. Henderson, and M. Stuttle. 2006. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: Generic slot-filling in the TALK in-car system. *EACL*.

Levesque, H. J., P. R. Cohen, and J. H. T. Nunes. 1990. On acting together. *AAAI*. Morgan Kaufmann.

Levin, E., R. Pieraccini, and W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8:11–23.

Levinson, S. C. 1983. *Conversational Analysis*, chapter 6. Cambridge University Press.

Levow, G.-A. 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. *COLING/ACL*.

Li, J., W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. 2016. Deep reinforcement learning for dialogue generation. *EMNLP*.

Li, M., J. Weston, and S. Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *NeurIPS19 Workshop on Conversational AI*.

Lin, Z., A. Madotto, J. Shin, P. Xu, and P. Fung. 2019. MoEL: Mixture of empathetic listeners. *EMNLP*.

Litman, D. J. 1985. *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues*. Ph.D. thesis, University of Rochester, Rochester, NY.

Litman, D. J. and J. Allen. 1987. A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11:163–200.

Litman, D. J., M. A. Walker, and M. Kearns. 1999. Automatic detection of poor speech recognition at the dialogue level. *ACL*.

Liu, H., J. Dacon, W. Fan, H. Liu, Z. Liu, and J. Tang. 2020. Does gender matter? Towards fairness in dialogue systems. *COLING*.

Lochbaum, K. E., B. J. Grosz, and C. L. Sidner. 2000. Discourse structure and intention recognition. In R. Dale, H. Moisl, and H. L. Somers, eds, *Handbook of Natural Language Processing*. Marcel Dekker.

Miller, S., R. J. Bobrow, R. Ingria, and R. Schwartz. 1994. Hidden understanding models of natural language. *ACL*.

Mrkšić, N., D. Ó Séaghdha, T.-H. Wen, B. Thomson, and S. Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. *ACL*.

Nagata, M. and T. Morimoto. 1994. First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 15:193–203.

Nayak, N., D. Hakkani-Tür, M. A. Walker, and L. P. Heck. 2017. To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. *INTERSPEECH*.

Neff, G. and P. Nagy. 2016. Talking to bots: Symbiotic agency and the case of Tay. *International Journal of Communication*, 10:4915–4931.

Nickerson, R. S. 1976. On conversational interaction with computers. *Proceedings of the ACM/SIGGRAPH workshop on User-oriented design of interactive graphics systems*.

Nielsen, J. 1992. The usability engineering life cycle. *IEEE Computer*, 25(3):12–22.

Norman, D. A. 1988. *The Design of Everyday Things*. Basic Books.

Ouyang, L., J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. 2022. Training language models to follow instructions with human feedback. *NeurIPS*, volume 35.

Paolino, J. 2017. Google Home vs Alexa: Two simple user experience design gestures that delighted a female user. *Medium*. Jan 4, 2017. `https://medium.com/startup-grind/google-home-vs-alexa-56e26f69ac77`.

Pearl, C. 2017. *Designing Voice User Interfaces: Principles of Conversational Experiences*. O'Reilly.

Perrault, C. R. and J. Allen. 1980. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3-4):167–182.

Pieraccini, R., E. Levin, and C.-H. Lee. 1991. Stochastic representation of conceptual structure in the ATIS task. *Speech and Natural Language Workshop*.

Rafailov, R., A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*.

Rashkin, H., E. M. Smith, M. Li, and Y.-L. Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. *ACL*.

Reeves, B. and C. Nass. 1996. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge University Press.

Ritter, A., C. Cherry, and B. Dolan. 2010. Unsupervised modeling of twitter conversations. *NAACL HLT*.

Roller, S., E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau, and J. Weston. 2021. Recipes for building an open-domain chatbot. *EACL*.

Roy, N., J. Pineau, and S. Thrun. 2000. Spoken dialogue management using probabilistic reasoning. *ACL*.

Sacks, H., E. A. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

Sag, I. A. and M. Y. Liberman. 1975. The intonational disambiguation of indirect speech acts. In *CLS-75*, 487–498. University of Chicago.

Schegloff, E. A. 1968. Sequencing in conversational openings. *American Anthropologist*, 70:1075–1095.

See, A., S. Roller, D. Kiela, and J. Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. *NAACL HLT*.

Shriberg, E., R. Bates, P. Taylor, A. Stolcke, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech (Special Issue on Prosody and Conversation)*, 41(3-4):439–487.

Singh, S. P., D. J. Litman, M. Kearns, and M. A. Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *JAIR*, 16:105–133.

Smith, V. L. and H. H. Clark. 1993. On the course of answering questions. *Journal of Memory and Language*, 32:25–38.

Stalnaker, R. C. 1978. Assertion. In P. Cole, ed., *Pragmatics: Syntax and Semantics Volume 9*, 315–332. Academic Press.

Stifelman, L. J., B. Arons, C. Schmandt, and E. A. Hulteen. 1993. VoiceNotes: A speech interface for a hand-held voice notetaker. *INTERCHI 1993*.

Stolcke, A., K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, M. Meteer, and C. Van Ess-Dykema. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–371.

Swerts, M., D. J. Litman, and J. Hirschberg. 2000. Corrections in spoken dialogue systems. *ICSLP*.

Ung, M., J. Xu, and Y.-L. Boureau. 2022. SaFeRDialogues: Taking feedback gracefully after conversational safety failures. *ACL*.

Wade, E., E. Shriberg, and P. J. Price. 1992. User behaviors affecting speech recognition. *ICSLP*.

Walker, M. A. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *JAIR*, 12:387–416.

Walker, M. A., J. C. Fromer, and S. S. Narayanan. 1998. Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email. *COLING/ACL*.

Weizenbaum, J. 1966. ELIZA – A computer program for the study of natural language communication between man and machine. *CACM*, 9(1):36–45.

Wilensky, R. 1983. *Planning and Understanding: A Computational Approach to Human Reasoning*. Addison-Wesley.

Williams, J. D., K. Asadi, and G. Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *ACL*.

Williams, J. D., A. Raux, and M. Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Williams, J. D. and S. J. Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(1):393–422.

Wittgenstein, L. 1953. *Philosophical Investigations. (Translated by Anscombe, G.E.M.).* Blackwell.

Wolf, M. J., K. W. Miller, and F. S. Grodzinsky. 2017. Why we should have seen that coming: Comments on Microsoft's Tay "experiment," and wider implications. *The ORBIT Journal*, 1(2):1–12.

Xu, J., D. Ju, M. Li, Y.-L. Boureau, J. Weston, and E. Dinan. 2020. Recipes for safety in open-domain chatbots. ArXiv preprint arXiv:2010.07079.

Yankelovich, N., G.-A. Levow, and M. Marx. 1995. Designing SpeechActs: Issues in speech user interfaces. *CHI-95*.

Young, S. J., M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.