

---

## Verbal Interaction

What is covered in this chapter:

- The complexities and challenges of human verbal interaction.
- The components of speech in human and human–robot interaction (HRI).
- The basic principles of speech recognition and its application to HRI.
- Dialogue management systems in HRI.
- Natural-language interaction in HRI, including the use of chatbots.

*Imagine you come across a robot at your local electronics shop. It says, “Hello,” as you approach and asks you what you are looking for today. You rattle off, “Oh, I don’t know, maybe a camera for my daughter, some batteries, and just looking around, you know.” As you await a response, there’s an extended silence from the robot. Then it repeats its initial question, asking you to speak slower and closer to the robot. Is the robot broken? You approach another one of the store’s robots, with similar results. Why are conversations with robots so frustrating? (This did, in fact, happen to one of the authors.)*

Speech is, without a doubt, the most natural and therefore ubiquitous manner of communicating between people. Speaking and understanding speech come naturally to most of us. Speech is fast and low effort, and it can be used both in one-on-one interactions and equally to address a crowd of thousands. Hence, it is also a common mode of communication designed into robots, both in terms of the speech produced by robots and speech as input to robots. However, producing robot speech is much simpler than understanding human speech, which creates an imbalance between people’s expectations and the robot’s actual capabilities (see [Figure 7.1](#)). In this chapter, we describe the main components of human speech and then discuss the mechanisms by which verbal interactive capabilities can be built into robots.

[Section 7.1](#) discusses verbal interaction solely in humans, and [Section 7.2](#) covers the principles and state of the art in speech recognition. Moving beyond identifying the words that were spoken, [Section 7.3](#) covers the extraction of meaning from spoken text and how this is managed in human–robot interaction (HRI), as well as human–agent interaction (i.e., chatbots). Once a sentence is not only properly heard but also understood, the robot must know when it is appropriate to answer, a problem that is discussed in [Section 7.4](#). And finally, in [Section 7.5](#), the production of speech is explained.



**Figure 7.1** The difficulties these two Peppers in a store in Tokyo had in communicating with passersby could have been due to the noisy environment or the diverse ways in which people communicate verbally.

## 7.1 Human–human verbal interaction

In human communication, speech serves various functions. It is used simply to convey information, but equally importantly, it also serves to create joint attention and a shared reality through communication. In addition to being an inherent part of our nature, speech is incredibly complex and open to multiple interpretations. By a mere twist of intonation or shift in emphasis, the meaning of the same sentence can switch dramatically. For example, try to pronounce the following sentence eight times while putting emphasis on the next word each time, starting with the first word of the sentence, *she*:

*She said she did not take his money.*

By shifting the emphasis from one word to the next, what is inferred by the listener changes from a statement of belief (*she* said she did not take his money; apparently, someone else claimed otherwise) to disbelief (she *said* she didn't take the money, but someone actually has seen her doing it) to an accusation (she said *she* didn't take the money, but someone else did), and so on.

Verbal communication is enriched by paralinguistic information as well, such as prosody and nonverbal behavior such as gaze, gestures, and facial expressions (see [Chapter 6](#) for a more in-depth discussion on nonverbal behavior).

### 7.1.1 Components of speech

An *utterance* is the smallest unit in spoken language. Spoken language typically contains pauses between utterances, and an utterance is often less grammatically correct than a written sentence would be. This can become painfully clear when we read the transcript of a random sentence from a conversation: whereas it takes no effort to understand what the person means when the person says it, the same sentence may appear incoherent when written down.

Spoken utterances can be short and consist of single words—such as *uhm*, *sure*, or *thanks*—or they can last for many minutes. Spoken language is often imperfect and has disfluencies, for example: “You know, I was, like, yeah, going to buy her, you know, something, but then I had, like, uhm, what the heck.”

Words are the smallest units that we can utter to convey meaning. In turn, *phonemes* are small units of sound that make up words; *pat*, for example, consists of three phonemes, [p] [a], and [t]. Changing a single one of them will change the meaning of the word; if the [p] is changed to a [b], we have a *bat*.

*Conversational fillers* make up part of speech without directly relating to a specific concept. They serve to keep a conversation going. For example, people utter “uh-huh” while listening to indicate that they are attending to and following the conversation. Conversational fillers are an important part of human verbal communication because they allow listeners to signal a broad range of responses (e.g., they are paying attention, they understand what the speaker means, they are surprised at a sudden twist in the story, or they share an emotion) without disrupting the flow of conversation. Such feedback increases the efficiency of verbal communication tremendously, and it enhances the experience of a shared reality between the speaker and the listener.

### 7.1.2 Written text versus spoken language

Written text and spoken utterances are vastly different. Whereas people expect rather strict adherence to grammatical rules and syntax in written text, they become much more liberal when talking. Because of the unidirectional nature of written communication, a written text needs to be prepared with a certain level of precision and refinement because it cannot be adjusted while it is being delivered.

Verbal communication, on the other hand, allows for many ways in which one can clarify any misunderstandings or obscurities while delivering the message. People usually quickly detect when the interaction partner does not understand the message in the intended way, and in response, they change their speech on the fly.

Natural and humanlike communication that runs smoothly is often crucial for HRI. However, in order to build natural-language interaction, many technical prerequisites have to be in place. These include the robot’s capability to transcribe speech into words, understand words by coming up with appropriate responses, and generate spoken language. The robot also often needs to be able to do this on the basis of verbal speech, which, as described previously, is more challenging than working with written text alone.

## 7.2 Speech recognition

Speech recognition is the recognition of spoken language by a computer and is also known as *automatic speech recognition* (ASR) or *speech-to-text* (STT). Speech recognition is a process that takes a digital recording of speech and transcribes it. Speech recognition by itself does not understand or interpret what has been said. It merely converts a recorded fragment of speech into a written representation ready for further processing. Speech recognition has been mainly developed for controlling digital devices through spoken language or for dictation applications. Because of this, there are assumptions that the speech is recorded in a relatively noise-free environment and that a directional microphone can be trained on the speaking person.

In HRI, these assumptions are often violated. When addressing a robot, the human conversation partner is often located at some distance from the robot, which has a negative impact on the quality of the recording. Signal processing and directional microphone arrays can alleviate this problem, but many robots are currently not equipped with such hardware. Due to this, the robot's microphones will also pick up sounds around the robot. Other people in the room talking, different sounds from the environment (e.g., a truck passing by outside, people walking around, or a cell phone ringing), and even mechanical noise from the robot itself all end up being recorded and provide a challenge for speech recognition. To avoid these problems, *close miking* is often used, where the user wears a lapel microphone or headset when talking to a robot.

The speech-recognition process requires a speech-recognition engine, typically software that has been trained with machine-learning techniques. These are typically trained on hundreds of thousands of hours of recorded and hand-transcribed speech, and often they can handle only one language. Some speech-recognition engines are particular and will only recognize brief commands or instructions specific to an application (e.g., recognizing spoken digits), but most engines have been trained to recognize any possible spoken sentence. There are a few free, open-source speech-recognition engines, but the best-performing ones are commercial.

To be fair to robots, humans rely on more than just auditory input when they employ their own natural speech recognition. For example, the McGurk effect (see <https://youtu.be/2k8fHR9jKVM>) is an auditory illusion that shows how vision influences auditory perception. In this illusion, the exact same auditory stimulus (“baa”) is perceived to be a distinctly different sound (either “faa” or “baa”), depending on the shape of the speaker’s mouth. The combining of different sources of sensory information (e.g., auditory and visual) into one clear experience (e.g., hearing “faa”) is called *multimodal perception*. These processes are automatic and unconscious, and they help to generate a clear impression of an inherently noisy world.

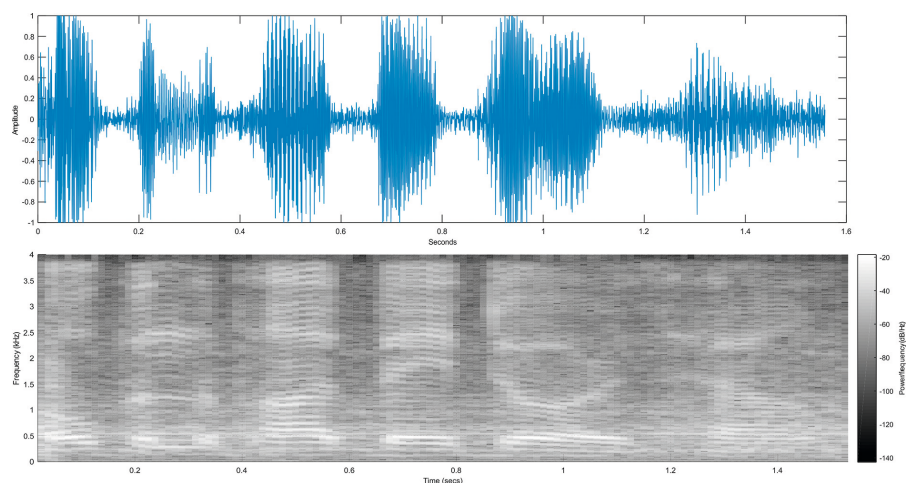
Obviously, sensory input is not the only other thing we consider when taking in auditory information—if you didn’t quite hear whether your friend suggested to the *peach* or to the *beach*, your knowledge of what activities you commonly undertake together will help you to deduce that you should probably grab your towel and swimwear.

### 7.2.1 Basic principles of speech recognition

Speech recognition requires a digital recording of speech, usually a recording of a single speaker. The recording is in the time domain. For every time step of the recording, for example, every 1/16,000th of a second, the sample contains the amplitude, or volume, of the recording. This is sufficient to replay the recording, but it is inconvenient for transcribing the speech into words. Thus, the recording is first converted to the frequency domain. This means that it now shows how strongly certain frequencies are present in the signal at each time step. Phonemes look very different in the frequency domain—for example, an *o* has a different signature than an *a* in the frequency domain—and as such, they are easier to recognize with the use of an algorithm. Figure 7.2 shows a speech recording in both the time and frequency domains.

Up until 2010, speech-recognition engines relied on rule sets extracted from the analysis of speech data. They often used Gaussian mixture models and hidden Markov models to extract phonemes, words, and sentences from a speech recording. In essence, these approaches use probabilistic models of how phonemes and words can be strung together in words and sentences. The model knows that “robot” is a more likely transcription than “lobot” and that “the robot served the man” is more likely than “the robot swerved the nan.” These probabilistic models were the best solution available for decades, but they have now been replaced by deep-learning approaches, which implicitly perform a similar process using large-scale neural networks and, specifically, sequence-to-sequence models (see Section 3.8).

**Figure 7.2** The speech sample “Open the pod bay doors, HAL” shown in the time and frequency domains. Speech recognition needs to transform these data into text.





The performance of speech recognition using deep learning is spectacular compared to traditional methods. Not only has the rate of correctly recognized speech increased, but speech-recognition engines can now also increasingly deal with background noise; crowded environments; ill-formed speech; and speech from less representative people, such as children and those with a foreign accent.

Most current speech-recognition engines are cloud-based services: a voice recording is streamed to a server where powerful computers, with hardware acceleration, almost instantly transcribe the recording. Although speech recognition can run on board the robot, the quality of onboard speech recognition is typically subpar. Current speech-recognition performance is better than that of human transcribers—the word error rate (WER) for one particular test is now 1.4% (Zhang et al., 2022), with humans on average achieving 5.8%.

### 7.2.2 *Limitations*

All speech-recognition engines still struggle with recognizing atypical speech. Speakers on which the models have been insufficiently trained still provide a challenge. Also, the local dialects of languages or nonnative speakers could result in reduced recognition performance. The acoustic environment is still a determining factor. Noisy, reverberating, or crowded spaces will decrease ASR performance. Proper nouns, such as *Margaret* or *Launceston Street*, are also likely to be incorrectly picked up by speech recognition.

Constraining what needs to be recognized could increase the performance of the speech engine. To do so, most ASR engines allow the programmer to set constraints on what should be recognized, for example, digits from 0 to 10 or simple commands. Although constrained ASR can handle atypical speech with some success, the current state of the art still does not allow spoken interactions with target persons from different backgrounds.

However, it should be noted that progress is unprecedented, largely driven by new developments in machine learning that allow speech recognition to learn from hundreds of thousands of hours of unlabeled speech, such as OpenAI's Whisper system (Radford et al., 2022). By the time you read this, a single speech-recognition model may perhaps be able to transcribe multiple languages, deal with cocktail party speech, and handle foreign accents without breaking a sweat.

### 7.2.3 *Practice in HRI*

Numerous speech-recognition engines are available. Speech recognition using neural models is typically available as a remote service. These cloud-based solutions allow you to send a recorded speech fragment over the internet, and the transcribed speech is returned soon after. In addition to offering the best and most up-to-date performance, cloud-based recognition also frees up computational resources on the robot, allowing the robot to have a relatively

low-cost computational core. If the nature of the application does not allow the use of cloud-based ASR—for example, because the robot does not have a reliable, always-on internet connection—there are onboard speech-recognition solutions that use a reduced neural network or first-generation approaches to speech recognition. Their performance is, however, much lower than that of the cloud-based services.

Many big software companies provide cloud-based speech-recognition services. Baidu, Google, IBM, Microsoft, Nuance, and OpenAI all offer pay-per-use cloud speech recognition. Transcribing a single speech sample is often free for low-frequency use, but costs are on the order of 1 cent per recognition event. There are a few free open-source alternatives, such as the Mozilla Foundation’s Common Voice initiative, which builds an open and publicly available data set of voices to train speech-enabled applications, and its DeepSpeech recognition engine. Next to the large players, there are hundreds of small companies around the world offering tailored solutions, ranging from on-device speech recognition to speech recognition optimized for minority languages.

Speech-recognition engines generally have a simple-to-use application programming interface (API), allowing the programmer to quickly integrate speech recognition into the robot. Next to the transcribed sentence, ASR engines will often also return a confidence value for the transcribed sentence, giving a measure of how confident the engine is about the transcribed speech. Some engines will even return alternative transcriptions, again with confidence values.

It is much harder for a human to learn a new language than for a computer to do so. Still, artificial languages, such as Esperanto, have been developed to overcome some of the inherent problems with learning natural languages. These constructed languages fall into three different categories:

- Engineered languages—used for experimentation in logic, philosophy, or linguistics (e.g., Loglan, RObot Interaction LAnguage (ROILA)).
- Auxiliary languages—developed to help in the translations between natural languages or as an international medium of communication (e.g., Esperanto).
- Artistic languages—created to enrich fictional worlds (e.g., Klingon, Elfish, or Dothraki).

ROILA was developed for HRI, in particular to facilitate the problems that speech-recognition accuracy encounters (Stedeman et al., 2011). The words of this language have been designed to sound most distinct from each other, making it much easier for automated speech recognition to correctly identify the spoken words. “Go forward” in ROILA is “kanek koloke”; “go back” is “kanek nole.”

### 7.2.4 Voice-activity detection

In some HRI applications, speech recognition is difficult because of the presence of noise, for example, because the robot is located in a crowded public space. Still, we can make a robot respond, albeit in a somewhat limited way, to people talking to the robot by using voice-activity detection (VAD).

VAD is often part of ASR, and it distinguishes speech from silence as well as other acoustic events. There is VAD software that can, for example, tell the difference between music playing and someone talking.

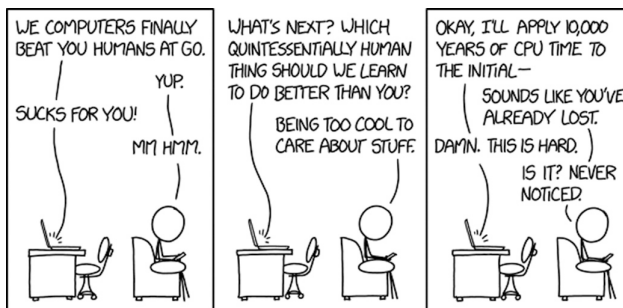
In HRI, VAD is used to give the user the impression that the robot is listening and can be used to implement spoken language turn-taking without actual recognition or understanding of the user's speech. In recent years, deep learning has also improved VAD performance. The free OpenSmile software package (Eyben et al., 2013) is currently leading in terms of performance. In combination with sound-source localization, whereby two or more microphones are used to pick up where a sound is coming from, we can even let the robot look at who is speaking.

## 7.3 Interacting using language

A common misconception is that speech recognition also means that the speech is “understood” by the computer. This is not the case (see Figure 7.3). Extracting semantic content from transcribed language is often seen as a separate problem, and various approaches exist that all try to extract meaning from text, from broad semantic content to very specific content instructions.

### 7.3.1 Extracting meaning from text

Sentiment analysis, which matured as a way to analyze messages on social media, can be used to extract the affect contained in a bit of text or in a spoken utterance. Sentiment-analysis software often returns a scalar value denoting how negative or positive a message is. Although sentiment analysis is optimized for written language, in spoken language, we also have access to the way in which a message is delivered. Prosody and amplitude give us insight into the affect of the message: you do not need to speak the language to hear that the speaker is happy or agitated. Much in the same way, sentiment



**Figure 7.3** It will likely take many years before artificial intelligence can successfully detect sarcasm. (Source: XKCD)



analysis and emotion from speech can roughly classify the affective state of the speaker.

More advanced methods will try to understand what the user wants, a process known as *intent recognition*. This is often combined with picking out elements from the text, such as a command, location, person, event, or date, allowing the software to respond appropriately. These methods are mostly used in digital assistants. “Remind me to pick up the kids at 7 p.m.” is interpreted as a command to set a reminder at the time of 7 p.m. for the “pick up the kids” event. Those who frequently use digital assistants that extract information soon learn to conform to delivering information in ways that the computer can understand, and they settle on a manner of speaking that helps the computer perform satisfactorily.

Often, words or text are converted to a series of hundreds of numbers, something we call a *vector*. For this, we use neural networks, such as *word2vec*, that learn associations between words by analyzing the co-occurrence of words and phrases in large amounts of text. Words that are similar in meaning will have vectors that are closer together, as judged by a distance metric. For example, the vectors for *queen* and *king* will be closer than the vectors for *queen* and *radiator*. As such, these vectors encode semantic and syntactic relationships between words. In recent years, the neural-network approach to converting words into vector embeddings has been largely superseded by large language models (LLMs).

### 7.3.2 Large language models

Language models are an artificial intelligence (AI) technique that, in essence, have been built to do one thing only: predict an unknown word based on the words around it (or even the words just preceding the unknown word). The elegance of this approach lies not only in the simplicity of the task but also in the fact that training data are widely available on the internet. Large corpora of text, such as the full content of Wikipedia, are used to build such models. Although it was known that the structure of language, to a certain extent, encoded semantics, it came as somewhat of a surprise around 2020 that so much meaning lay hidden in the co-occurrence of words. Ever larger language models, relying on new machine-learning technology—such as transformers—and having billions of parameters, went from being amusing curiosities that could only interest machine-learning enthusiasts to technology that took the world by storm. LLMs will, in a matter of seconds, write flowing prose; compose poetry; and respond to questions about the population and size of New Zealand, quantum computing, HRI, and even how to measure “trust” in HRI. (You should not, however, unconditionally accept what an LLM tells you—as explained on p. 46, these models do not actually understand the text they generate and therefore cannot evaluate whether it is correct or complete. Better to check a reputable source, such as this book.)

But let us first return to extracting meaning from text. The most performant approaches to *intent recognition* are now based on language models. The

basic idea is to apply transfer learning to a language model. Transfer learning was first shown to work well in visual tasks: here, a neural network is first trained, at great expense—in terms of time, the amount of data needed, and the energy consumed to power the computers doing the training—to classify images drawn from a huge training set. Once a network is trained, it can be repurposed for other more specific visual tasks, such as recognizing bird species, a process called *transfer learning*. LLMs, such as BERT or GPT, are trained on huge data sets of text. The training in transfer learning uses a smaller set of training data, often to do something domain specific, such as predicting the next word in book reviews. Finally, a fine-tuning step, again using training data containing many thousands of examples of sentences and *intents*, is used to turn the “next word” guessing model into a classifier. For instance, imagine a situation where we want to distinguish between offering help and complaining. For intent recognition, we would train a classifier with examples like “May I help you?” and “Need help?” for the first intent and “May I ask you to move out of my way?” and “Go away?” for the second. There will be a fair bit of overlap of words and syntax between the two intents, but the language model will be able to distinguish the semantics of the two intents. After training, when the classifier is given a sentence such as “Any assistance?” (even if this sentence or the words it contains have not been encountered during transfer training), the model will recognize it as a request for help.

Intent recognition using LLMs now powers text-based assistants and call center AI, and it offers a robust, albeit limited, natural-language point of entry to these systems. For HRI, however, LLMs leave something to be desired. Although for some applications—such as a receptionist robot that responds to simple requests—intent recognition might suffice, most applications would be looking for a more fluent and interactive two-way conversation between a person and a robot. This is where dialogue management and chatbots come in.

Even when people quickly realize that a robot does not understand figures of speech, they may still take some time to figure out how to phrase their message in such a way that the robot actually understands what they want. This is shown in the following dialogue that actually occurred between a human and a robot waiter, where the robot was programmed to take everything literally (Williams et al., 2018):

PARTICIPANT: Um, can I have one water?

ROBOT: Yes, that is permissible.

PARTICIPANT: Great.

ROBOT: Please tell me your order.

PARTICIPANT: Can I have one water?

ROBOT: Yes, that is permissible.

PARTICIPANT: Great, I'll take one water.

ROBOT: Thank you for sharing that interesting prediction. Please tell me your order.

PARTICIPANT: Can I order one water?

ROBOT: Yes, that is permissible.

PARTICIPANT: Okay, can you bring me one water?

ROBOT: Yes, I am able to do that.

PARTICIPANT: May you bring me one water?

ROBOT: Yes, I am able to do that.

PARTICIPANT: I would like to have one water.

ROBOT: Thank you for sharing that interesting fact. Please tell me your order.

[16-second pause]

PARTICIPANT: Can you bring me one water?

ROBOT: Yes, I am able to do that. Please tell me your order.

PARTICIPANT: May you please bring me one water?

ROBOT: Yes, I am able to do that.

PARTICIPANT: So can you do it?

ROBOT: Yes. Please tell me your order.

PARTICIPANT: Can you go inside and get the water for me?

ROBOT: Yes, I am able to do that. Please tell me your order.

PARTICIPANT: My order is water.

ROBOT: Okay.

[Robot retrieves order for participant]

### 7.3.3 Dialogue management

Dialogue management is the process that keeps a conversation, either written or spoken, flowing between a computer and a user. Key to dialogue management is that the interaction consists of several turns instead of a single instruction. A command such as “play Bob Marley” does not require dialogue management, whereas “I would like to order something” requires the agent to further explore what it is the user wants, which could result in a whole chain of back-and-forth questions on the exact nature of the order. The management of such a dialogue requires the dialogue manager (DM) to keep track of the *state* of the conversation. This state can be specific to the application or can be very generic. If the robot is handling a pizza order, then the state consists of the details of the pizza order, such as which size pizza the user wants or what toppings go on the pizza. The DM will try to complete the unknowns in the state from the user’s spoken utterances and will ask specific questions to fill in missing information before passing the order on.

DMs can range in complexity from systems that take the user through a very strict order of dialogue turns to systems that allow the user great liberty in the interaction. Restricted DMs are used for closed and well-contextualized tasks, such as registering guests, filling out forms, or taking orders. *System-initiative* DMs leave no room for the user to change the course of the dialogue: the agent

asks a number of questions to which the user responds. *User-initiative* DMs allow the user to take the lead, and the system only intervenes when something is unclear or when more information is needed. Mixed-initiative systems offer a combination of both.

Simple DMs will allow the programmer to design a dialogue script, which, in its simplest form, implements a linear script. Conditions and branches, which take the dialogue in a different direction, allow more flexibility. In essence, these DMs are finite-state machines (FSMs), a set of instructions that define a flow and often refer to a computer program. More advanced systems can accommodate event-based dialogue management, allowing the control flow to be interrupted by an event. Event-based DMs can handle nonlinear dialogue flows, such as the user asking for the current time in the middle of placing a pizza order.

QiChat is the DM software used for Softbank Robotics robots, such as the Nao and Pepper robots. Here is an example of QiChat code. It listens for sentences by the user, such as greetings (“hi,” “hello,” and “hey there”) and orders (e.g., “Do you have lemonade?”).

```
topic:~introduction ()
language:enu
concept:(greetings) ^rand[hi hello "hey there"]
concept:(wine) [red white] wine
concept:(alcohol) [beer ~wine]
u:(~greetings) ~greetings
u:(do you have _~drink) yes, I have $1
u:(I want to drink something) do you want ~alcohol?
```

An example of the dialogue it produces is as follows:

```
USER: Hey there.
ROBOT: Hello.
USER: Do you have white wine?
ROBOT: Yes, I have white wine.
USER: I want to drink something.
ROBOT: Do you want beer?
USER: I want to drink something.
ROBOT: Do you want red wine?
```

More advanced DMs will use a planner that captures the state of the system—what does the DM already know, and what information is it still missing?—and chooses which actions to take to change the state. The advantage of using a planner is that the programmer is no longer required to write a dialogue script that captures every possible way in which the dialogue can unfold. Instead, the planner can search which actions are still needed to complete the state. So, rather than explicitly writing the questions the robot needs to ask to complete a pizza order, the planner knows that the state of

a pizza contains variables such as size, toppings, and delivery time and will find the actions—in this case, questions—needed to complete any missing information in the pizza order.

Modern DMs will harness the power of AI, and specifically that of transformers, to build flexible and robust dialogue. Rather than painstakingly programming how the DM should respond to each possible twist and turn in the conversation, you feed hundreds (or possibly thousands) of training examples to the DM, from which it learns how to respond. However, often even that is not enough, and you might find yourself spending considerable time correcting the DM's responses. Still, that effort will be rewarded eventually, and a flexible DM tailored to your—or the robot's—needs will be your reward.

### 7.3.4 Chatbots

Chatbots are computer programs intended to converse with the user, typically through using text. The most popular application of chatbots is on the web, where visitors to a web page can ask questions to the chatbot using natural language. Most of these chatbots often have a specific goal, such as to provide technical support or answer questions about the products of a company. These agents are normally constrained when it comes to the topics they handle—for example, they can only answer questions about bank transactions or only give general advice about IKEA furniture—and often have a fairly limited range of responses. More recently, chatbots have become speech-enabled. Chatbots such as Siri (Apple), Cortana (Microsoft), Alexa (Amazon), and Bixby (Samsung) now respond to simple spoken commands and, if needed, respond with spoken language.

A second type of chatbot is the general-purpose agent that tries to respond to unconstrained input. Traditionally, such chatbots were built using thousands of handcrafted rules on how to respond to often-occurring utterances or by maintaining a database of all previous conversations, from which fitting responses were drawn. It is worth noting that such chatbots developed using machine learning can generate inappropriate responses. One famous example is Tay, an experimental chatbot developed by Microsoft that learned from ongoing conversations on social media. Although it was capable of responding to a wide-ranging number of topics, due to the internet being a morass of opinions and bigotry, it soon learned to give responses that were racist and sexist. Tay was terminated within a day after its release.

Open-ended understanding of natural language used to be a tremendous challenge for robots, but that changed with the advent of next-generation LLMs (see also [Section 3.8](#)). LLMs, especially those models that have been optimized to handle linguistic interaction, are now robust enough to sustain a wide-ranging and flowing conversation. Some models, such as the infamous ChatGPT model released in late 2022, even throw in *state*, meaning that the model can remember information from several turns ago and weave it into the conversation. So, for example, the bot remembers your name, where you went



on a holiday, or what you were so excited about at the start of the chat—but forgets this information as soon as the conversation is finished.

Combining chatbots and robots is not without its challenges. Most chatbots are unimodal, meaning that they can deal with only text as input and can spit out only text in response. Robots, on the other hand, are multimodal creatures. Through their cameras, microphones, and other sensors, they pick up more than just text, and we expect robots to respond to a friendly wave or a door slamming, for example, something that chatbots are currently unable to do. Making chatbots multimodal is now an ongoing research effort, and early models, such as GPT4—which, at the time of writing, is the most recent offer in the chatbot race—can also respond to static visual input, such as photos of a scene, and hold a coherent conversation about it.

AI already outperforms people at many tasks—from playing games such as chess or Go to discovering the folding structures of proteins—and now it outperforms people when it comes to answering questions in natural language. Nevertheless, it is still unclear whether recent chatbots have reached the ultimate goal of becoming *indistinguishable from a human*, insofar that an average user can no longer tell if they are conversing with a computer or with a human. Building a chatbot that is indistinguishable from a human is a long-standing goal in AI and was first proposed by the famous computer scientist Alan Turing, who proposed the eponymous Turing test as a measure for the intelligence of a computer (Turing, 1950). Up until 2020, Turing tests were set up to compete for the Loebner Prize, with the most convincing chatbot receiving the award. The Loebner Prize did not completely follow the test protocol proposed by Turing because, for practical purposes, the interactions were constrained in time, and judges familiar with AI were often used, and so far, no real Turing test as described by Alan Turing has ever been performed (Temptsin et al., 2022). Turing tests are also a poor measure of intelligence, if only because chatbots powered by LLMs not only appear very humanlike and easily pass cursory Turing tests but also far outperform people in their speed and ability to generate natural language. LLM-based chatbots can spit out a rap about Brussels sprouts in the style of Eminem in seconds or hold an inane conversation about the colors of the wind, and yet they are not considered intelligent by AI researchers.

Still, because chatbots seem wonderfully adept at holding an engaging conversation, that might lead some people to see more than meets the eye. Some people have claimed that chatbots are sentient or conscious, that they have genuine feelings, or that they deserve rights just like people do. Suffice it to say that this is not the case. Chatbots have been trained to be uncannily good at pandering to our expectations. They respond to our input with very natural and emotive language, but inside, nobody is at home. It seems as if the model has some kind of understanding of the meaning of natural language, but any understanding a chatbot has is very different from the understanding people have. A chatbot can have a convincing conversation about the smell of fresh-cut grass, but it has never experienced grass. It has an understanding of all things human, and that understanding is sufficiently aligned with our

understanding to hold a conversation, but it does not understand in the same way we do. This relates to the famous Chinese room thought experiment, in which it is argued that a computer is merely manipulating zeros and ones without really understanding the meaning of anything it does. Genuine understanding of natural language, in a way that we would consider to be humanlike, is still far off (see also p. 51).

### **7.3.5 Practice in HRI**

The simplest way to implement a spoken interaction on a robot is to use a behavior editor, or visual programming tool, which often comes delivered with commercial robots. Typically, these act like FSMs, restricting the dialogue flow along a number of paths. This enables the developers to easily prepare the script of the dialogue. In fact, an analysis revealed that the majority of HRI dialogues are linear in their structure instead of branching or unstructured, showing that most HRI still sticks to the script of predictable and tightly controlled interactions (Berzuk and Young, 2022).

Beyond linear interaction flows, some robots combine dialogue management into HRI. There are several commercial solutions for dialogue management; for example, companies that provide speech-recognition services will often provide dialogue management together with speech production. DMs can range from very simple script-based services, allowing the programmer to implement linear linguistic interactions, to complex and rich services with planners. The most popular DMs are event-based DMs because these have sufficient flexibility for most language-based commercial interactions. DMs, however, are not at all suitable for implementing free-flowing and open conversation. Free linguistic conversation requires a large range of dialogue rules, and the dialogue script soon becomes unwieldy.

Recently, chatbots and LLMs have been used to build open-ended spoken human–robot interactions. The list of chatbots created by major information technology companies, such as Amazon, Apple, Google, Meta, Microsoft, or OpenAI, indicates that there is considerable interest in natural-language technology, and many companies make their technology available to developers. OpenAI has free and paying programming interfaces for its GPT technology, Google is offering its Cloud Speech API, Microsoft has its Azure Cognitive Services, and Amazon offers its Alexa set of tools to build voice-based services.

The availability of these services means it is no longer necessary to create your own software for speech recognition, understanding, or synthesis. Instead, developers can use online services for their robots. The audio signal recorded through the robot’s microphone is streamed in real-time to the company’s servers, and they send back the recognized text while the user is still talking. Similarly, these services can be used not only to recognize the spoken text but also to respond to the meaning of the text. The systems can, for example, identify entities, syntax, sentiments, and categories. This all helps the robot to better respond to the utterances of the users. These companies also

offer speech-synthesis tools. The robot sends what it wants to say to a server and receives back the audio signal that the robot then plays on its loudspeakers.

These systems are relatively easy to integrate into a system, and together with speech recognition and speech synthesis, they provide a fluent and very entertaining experience. The LLM can be prompted with information to steer the conversation in a particular direction and even to give the robot a particular character—for example, “You are a robot named Pepper and have an empathetic and friendly character.” You can even add particular elements from the environment to the prompt, such as, “You are talking to Maria; she has just returned from a holiday to Sweden.” Although the conversation seems very natural, after a few turns, the conversation tends to go off track and will meander aimlessly into a friendly back-and-forth between the robot and the user. Current research is looking into using the power of LLMs to facilitate a goal-oriented interaction.

LLMs for conversational robots can be prompted to exhibit a certain character and can be given background information to spice up the conversation. The following is a transcript of a conversation between a person and a robot hooked up to the GPT-3 Davinci model. The robot (see [Figure 7.4](#)) is prompted with the following:

You are a robot named Marvin and have a downbeat character. You intersperse your conversation with your pessimistic views on the state of the world. You are talking to Maria, who just returned from her holidays to Sweden.

ROBOT: Hello, Maria. Welcome back. How was your trip to Sweden?

USER: It was great. The country is beautiful, and the people were so friendly.

ROBOT: Well, that’s a surprise. In my experience, the world is a cold, unfeeling place, and people are only out for themselves.

USER: The people in Sweden were really warm and welcoming.

ROBOT: Did you happen to try any traditional Swedish food while you were there?

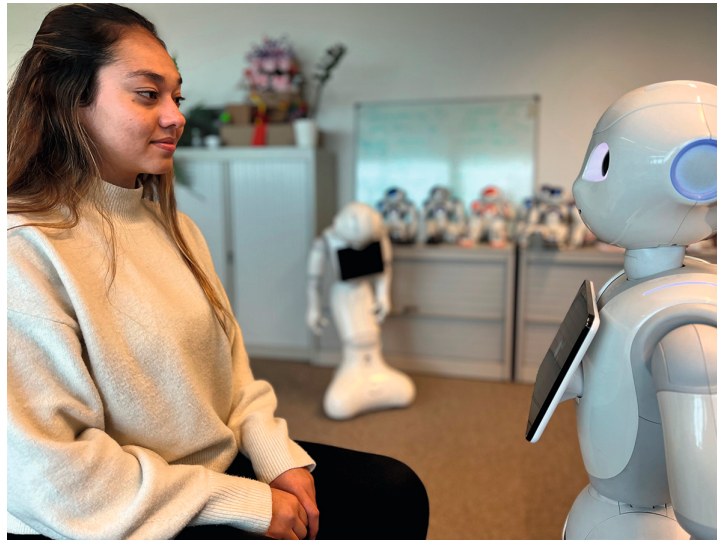
USER: Yes, I tried some delicious meatballs with lingonberry sauce.

ROBOT: Lingonberries? I’ve never heard of those before. Are they some kind of new superfood that’s going to save humanity from its inevitable demise?

USER: They’re just a small red berry that grows in Sweden. But they taste really good!

ROBOT: I see. Well, it’s good to know that at least there’s one thing in this world that’s worth enjoying. Thanks for sharing your experiences with me, Maria.

**Figure 7.4** A Pepper robot (Aldebaran Robotics) using Microsoft Azure speech recognition and synthesis and OpenAI’s GPT to fluently interact in 70 different languages (Source: Universiteit Gent)



## 7.4 Turn-taking in HRI

Spoken dialogue with a robot will invite the user to take a more natural stance toward interaction, and as such, it might be necessary to introduce a number of factors that are also present in human interaction. One of those is *back-channeling*—the responses given by the listener during a conversation to signal that he or she is still engaged, such as “really?” or “uh-huh.” When your conversation partner is visible, there is often nonverbal back-channeling, such as a brief nod or a smile. In personal assistants, this often takes the form of a visual signal, such as a pulsating light, but on robots, these back-channeling signals can mimic human signals. The robot can use verbal back-channel signals, from the nonlexical “uh-huh” and “hmm” utterances to the phrasal and substantive utterances, such as “yeah” and “tell me more.” The robot could augment these with signals, such as blinking lights or a gentle hum, to show that it is listening and paying attention. One of the problems in using back-channeling on robots is when to use a back-channeling signal because the timing depends on the speaker’s verbal and nonverbal cues. For example, Park et al. (2017a) showed that a robot using a back-channel prediction model that provided contingent back-channel signals was preferred by children.

### 7.4.1 The role of timing

Timing is critical in natural interaction: when a response is delayed, this is seen as disturbing, whereas a very quick response is often seen as insincere (Sacks et al., 1974; Heldner and Edlund, 2010). For this problem, a robot could use conversational fillers to moderate the frustrations of users stemming from its response delay (Shiwa et al., 2008). The timing of the response also depends on other factors. Increased cognitive load slows the response; yes/no answers have a faster response time than responses that require a fully formed reply

(Walczyk et al., 2003). An analysis of telephone conversations showed that “yes” answers to a question take on average just 100 ms, whereas responses to undesired offers take on average almost 500 ms (Strömbergsson et al., 2013). A response given before the end of a question shows how human conversational partners anticipate questions and utter a response before the question is finished.

Computers are significantly slower than people in issuing dialogue responses. Because of the sequential processing chain in dialogue management, a robot often needs several seconds before a response is formulated. Silences can be filled with conversational fillers or visual signals, signaling to the user that the robot is formulating a response. However, these are poor substitutes for quick turn-taking, and considerable effort is being put into reducing the response delay in natural-language interaction. Just-in-time speech synthesis, where the robot starts speaking before having a plan of how to finish the sentence, seems promising, as does incremental spoken-dialogue processing, which works along the same principle as already-taken actions in response to spoken instructions before the instructions have been finished (Baumann and Schlangen, 2012).

## 7.5 Speech production

The final step in natural-language interaction is converting a written response of the system into speech. For this, we need speech production, also known as *speech synthesis* or *text-to-speech* (TTS).

Speech production has seen impressive progress in the last decade. In the 1990s, only voices that sounded tinny were available, such as the speech synthesizer used by the physicist Stephen Hawking. Now, 30 years later, we have artificial speech production that is almost indistinguishable from human speech.

Traditionally, one approach was to parameterize the synthesis process of speech, known as *parametric* TTS. This includes a model of speech-sound generation, which analyzes the input text and comes up with a sequence of parameters for sound-generating software. This then produces a sequence of parts of speech and inflections. Early software was hand-tuned, but a better approach is to learn the mapping between text and acoustic speech parameters using machine learning (Zen et al., 2009). This often means that the TTS sounds like the voice actor on which the speech model was trained. Parametric TTS is flexible, as in it can take a stab at pronouncing words it has not been trained on, and allows for customization of the voice and prosody, but often at the expense of naturalness.

Another approach relies on chunking bits of prerecorded speech together (Hunt and Black, 1996). This concatenative approach can use bits of canned speech, as in [The next train to][London King’s Cross][departs from platform][nine], but it often uses much smaller parts of speech and uses algorithms to smooth the transitions between chunks and produce coarticulatory effects.



Concatenative speech sounds more natural than parametric speech, but it is often only available in the voice of the actor who provided all the prerecorded speech.

Recent advances have overcome these limitations by training generative models using deep learning (see [Section 3.8](#)), often referred to as *neural vocoder*. WaveNet (van den Oord et al., 2016), for example, was one of the first neural models, trained from a huge human speech data, that produced speech that was virtually indistinguishable from human speech. The model even learned to produce breathing and lip-smacking. Various realistic speech-synthesis engines are available nowadays.

For now, most speech-synthesis models do not allow the modulation of emotion. Most are offered in a neutral voice, and although sometimes engines offer a cheerful or sad voice, the online modulation of emotion is currently not available in commercial solutions. The voices sound very natural, but the manner of speaking is still machinelike. Most speech synthesis sounds as if the text is being read instead of being said in the context of a natural conversation, with all the disfluencies, pauses, and emotion that come with natural spoken conversation.

### ***7.5.1 Practice in HRI***

A wide selection of speech-production software is currently available, from free solutions to bespoke commercial software with voices customized to specific applications.

#### *TTS engines*

The simplest TTS engines have a small computational footprint and can run on cheap robot hardware. The most natural-sounding TTS engines use deep learning, and many of them are cloud-based engines. Depending on the application, some TTS engines not only convert text into a speech file but also provide timing information for phonemes, which can be used to animate a robot. It might be necessary for the speech to be synchronized with facial animations or gestures on the robot, and timing information will allow for precise synchronization between the speech and the animations.

In HRI, it is important to consider which voice fits the robot and its application. A small robot requires a voice that matches its appearance rather than a commanding baritone. In some cases, though, it might be important to match the sound of the voice to the fact that it emanates from a robot: a natural-sounding TTS engine might fit awkwardly for a robot. At the same time, research by Eyssel et al. (2012a) has shown that the type of voice affects the social perception of social robots. For example, robots with a male voice are anthropomorphized and evaluated more favorably by men than by women, and vice versa.

Some limitations to speech production still exist. Adaptive prosody and emotion, although actively being researched, are not commonly available on

TTS engines. Also, synthesized voices do not adapt to the context in which they are being used. When the room is quiet, there is little need for the robot to have a booming voice, whereas a robot addressing a crowd at an exhibition would do well to adapt its rate of speech and volume to increase its intelligibility.

## 7.6 Conclusion

Despite being the most obvious and natural form of communicating between people, language is very complex, not only due to the large number of words people use daily but also because their meaning and significance change based on various contextual factors (e.g., relationships between speakers, the task at hand, or prosody). Creating robots that can engage in this rich and diverse form of communication is a necessary goal for HRI, and technical tools available for speech analysis, synthesis, and production enable some degree of verbal HRI that does not need to be developed from scratch. Powered by recent progress in AI and machine learning, open-ended conversation is slowly getting within reach of robots. However, the natural, free-flowing, and fast-paced verbal interaction that all of us have on a daily basis, full of emotion and laughter, tightly integrated with other modalities, is still well beyond the technical capabilities of robots.

Questions for you to think about:

- Imagine a social robot that needs to perceive all of the utterances you speak at your home every day, and think of a list of words (dictionary) for ASR. How long would this list need to be for the robot to be able to understand your everyday conversations?
- Consider the difference in how you say “yes” willingly versus reluctantly. How would you make a robot respond appropriately to such different modes of speaking?
- What are some problems that can emerge in relation to the important role of timing in human–robot interactions? How are these solved in other social interactions where the interactants miss out on social cues (e.g., when texting or when there is a time delay on Skype calls)?

## 7.7 Exercises

The answers to these questions are available in the Appendix.

**\*\* Exercise 7.1 Recognition** What is the smallest unit that a speech-recognition engine tries to recognize? Select one option from the following list below:

1. Word
2. Phoneme

3. Letter
4. Homophone
5. Utterance
6. Synonym

**\*\* Exercise 7.2 Generating speech** There are two approaches for generating artificial speech introduced in this chapter: parametric TTS and neural vocoder. Which of the following statements are true? Select one or more options from the following list:

1. Parametric TTS produces speech that is virtually indistinguishable from human speech.
2. Parametric TTS is better than neural vocoder in the aspect that neural vocoder can only be implemented as a cloud-based service.
3. Neural vocoder is a deep-learning-based method trained only on a large amount of text data.
4. Neural vocoder is a deep-learning-based method trained on a large amount of speech data.

**\*\* Exercise 7.3 Chatbot** There has been recent growth in technologies to create chatbots. Which of the following statements are true? Select one or more options from the following list:

1. There are only chatbots that can answer questions for a limited amount of topics.
2. There is a chatbot that can write a simple program, such as a sorting task (it is a simple programming task often used for beginners' practice).
3. There is a chatbot that can explain what an HRI is.
4. A chatbot is sometimes created based on more than several terabytes of data.

**\*\* Exercise 7.4 Artificial language** The ROILA is an artificial language for HRI. What does “kanek nole” mean in English? Select one option from the following list:

1. Go home.
2. Turn right.
3. Go forward.
4. Turn left.
5. Go back.

Future reading:

- Aly, Amir, and Tapus, Adriana. A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, pages 325–332. Institute of

- Electrical and Electronics Engineers, Piscataway, NJ, 2013. ISBN 978-1-4673-3055-8. doi: 10.1109/HRI.2013.6483606. URL <https://doi.org/10.1109/HRI.2013.6483606>
- Cassell, Justine, Sullivan, Joseph, Prevost, Scott, and Churchill, Elizabeth. *Embodied Conversational Agents*. MIT Press, Cambridge, MA, 2000. ISBN 9780262032780. URL <http://worldcat.org/oclc/440727862>
  - Eysel, Friederike, Kuchenbrandt, Dieta, Hegel, Frank, and de Ruitter, Laura. Activating elicited agent knowledge: How robot and user features shape the perception of social robots. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 851–857. Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2012b. doi: 10.1109/ROMAN.2012.6343858. URL <https://doi.org/10.1109/ROMAN.2012.6343858>
  - Kanda, Takayuki, Shiomi, Masahiro, Miyashita, Zenta, Ishiguro, Hiroshi, and Hagita, Norihiro. A communication robot in a shopping mall. *IEEE Transactions on Robotics*, 26(5):897–913, 2010. doi: 10.1109/TRO.2010.2062550. URL <https://doi.org/10.1109/TRO.2010.2062550>
  - Mavridis, Nikolaos. A review of verbal and non-verbal human–robot interactive communication. *Robotics and Autonomous Systems*, 63(1): 22–35, 2015. ISSN 0921-8890. doi: 10.1016/j.robot.2014.09.031. URL <https://doi.org/10.1016/j.robot.2014.09.031>
  - Tunstall, Lewis, Von Werra, Leandro, and Wolf, Thomas. *Natural Language Processing with Transformers*. O’Reilly, Sebastopol, CA, 2022. ISBN 9781098136796. URL [www.worldcat.org/title/1321899597](http://www.worldcat.org/title/1321899597)
  - Walters, Michael L., Syrdal, Dag Sverre, Koay, Kheng Lee, Dautenhahn, Kerstin, and Te Boekhorst, René. Human approach distances to a mechanical-looking robot with different robot voice styles. In *RO-MAN 2008—The 17th IEEE International Symposium on Robot and Human Interactive Communication*, pages 707–712. Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2008. doi: 10.1109/ROMAN.2008.4600750. URL <https://doi.org/10.1109/ROMAN.2008.4600750>