

# Design a Bot. Today.

Conversational User Experience Guide





## **Overview Guide**

#### **INTRODUCTION**

#### **CONVERSATIONAL UX**

Topic 01. What is a conversational user experience?

Topic 02. Benefits of a conversational experience

Topic 03. <u>Creating a responsible conversational experience</u>

Topic 04. Types of conversational experiences

#### DESIGNING THE IDEAL EXPERIENCE

Topic 05. <u>Identifying key scenarios</u>

Topic 06. <u>Understanding and identifying customer intents</u>

Topic 07. Disambiguating customer intent

#### PRACTICAL TIPS

Topic 08. Queries, utterances, and triggers

Topic 09. Graceful fallbacks and hand-offs

Topic 10. Error handling

Topic 11. **Ending conversations** 

#### TOOLS

Worksheet. **Envisioning your bot** 





## Introduction

Welcome , fellow conversation designers! You're in great company. Conversational designs—bots, virtual agents, digital assistants, and all other text-based dialog interfaces—offer flexible, accessible, and powerful ways to connect with customers, employees, and one another. Whether you're a novice designer or a veteran developer, these topics offer insights to help you craft an effective, responsible, inclusive, and, we hope, delightful experience. The authors and designers of this guidance are drawing from combined decades of experience building and deploying conversational UI for a variety of bots, virtual agents, and other conversational experience projects including Cortana, the Microsoft Virtual Agent, Bot Framework Templates, Personality Chat, and others.

This guidance is divided loosely into a few different sections. The first four topics introduce conversational design and explore ways to approach the process ethically and inclusively. Topics 5-7, along with our brainstorming worksheet, help you design your ideal experience. Finally, topics 8-11 offer practical development tips for building the experience you've designed. You can jump around and choose only the topics that address your immediate need, or go top-to-bottom in order.

A note on terminology: topic four explores several different kinds of conversational experiences, including bots, virtual agents, and digital assistants. Within the rest of the guide, we use those terms relatively interchangeably because the principles of development in this guidance apply interchangeably to all of those design types, but we recognize there are distinctions in the industry between those and other conversational designs, and that those terms are sometimes used differently in the industry. Our intention is to offer guidance that will help with most text-based conversational experiences, regardless of their intent.

## **CONVERSATIONAL UX**





## 01. What is a conversational user experience?

We define conversational user experience (CUX) as a modality of interaction that's based on natural language. When interacting with each other, human beings use conversation to communicate ideas, concepts, data, and emotional information. CUX allows us to interact with our devices, apps, and digital services the way we communicate with each other, using phrasing and syntax via voice and text or chat that come naturally.

Natural Supportive Conversation System

Other modalities can burden users with the task of learning interaction behaviors that are meaningful to the system: the syntax of a command line, the information architecture of a graphical user interface, or the touch affordances of a device. CUX turns the tables. Instead of users having to learn the system, it's the system that learns. It learns what we teach it about human language—patterns of speech, colloquialisms, chit-chat, even abusive words—so that it can respond appropriately.

#### **CONVERSATION IS PERSONAL**

There's more to developing a conversational experience than crafting dialogue. The question of what constitutes an appropriate response—and the choice of who gets to decide what's appropriate—are ethical concerns and fundamental to the design process. More than other modalities of interaction, conversation is especially personal, ingrained with human qualities of emotion, agency, and personality. Conversational experiences that fail to address these human traits risk being unsatisfying at best and slanderous at worst.

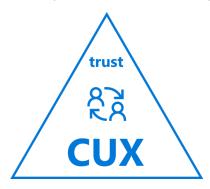
Take personality, for example. When a device, app, or service responds to user input, the user begins to assign a personality to it, even when the interaction isn't actually conversational. This phenomenon of assigning a personality is more pronounced when the user is communicating in their natural language, whether in text or in their speaking voice. In fact, our latest research tells us that users become more attached to devices when they interact using natural language. It's therefore in the best interest of the CUX designer to take into account the social and emotional implications of establishing a personal relationship between the user and the device, app, or service. CUX designers have a responsibility to honor the emotional state of the human on the other end of the conversation.

#### THE POWER OF BEING UNDERSTOOD

When a system responds intelligently to natural language, it's not only signaling that it received a message or command, it's communicating that it comprehends the user's unique form of human expression. The user feels heard. Going further, if the system's response includes emotional nuance relevant to the user's cues, the user not only feels heard, but they feel



understood as well. In demonstrating intelligence and empathy, CUX lays the groundwork for building a relationship of trust with every person who uses the system.



CUX can be multi-modal—employing text or voice, with or without visual, auditory, or touch-enabled components. But fundamentally, CUX is human language. In 2016, Microsoft CEO Satya Nadella announced, "Human language is the new UI layer." CUX lets you harness the power of words to build trust and forge deep connections with your users.



## 02. Benefits of a conversational experience

The ability to use natural language to interact with a device, app, or feature is taking its place alongside typing, tapping, and touch as a critical UX option. For some situations, in fact, a conversational experience can be the most efficient, accessible, intuitive, and empathetic way to accomplish a task. It's can also offer a powerful way to help you build trust with your users. Here are some of the ways that conversation excels as a user experience.



#### **EFFICIENCY**

For some features, the fastest way to get a task done is to use natural language. Take, for example, a simple task like setting an alarm. Traditionally, this requires users to find and then open an app, manually select the time, choose AM or PM, and then remember to save it—up to 8 clicks on some apps. With conversational experiences, it's as simple as saying "Set an alarm for 8 AM."

Other command-and-control tasks, like adjusting the volume, turning on Bluetooth, setting reminders, or looking up quick facts like weather or traffic can be simple hands-free, eyes-free experiences when built properly. With digital assistants, it's crucial to consider your user's most common questions, and offer simple answers that helps to meet their expectations quickly and efficiently. This will help them achieve their goals.

#### **ACCESSIBILITY**

A disability is a mismatch between a person and their environment, and it can be situational, temporary, or permanent. We've reached a beautiful moment in user experience (UX) design where we no longer have to sacrifice accessibility to the other demands of a design process. Voice-driven CUX can be the bridge that matches people's abilities with their environments, when other modalities, such as visual or tactile, aren't an option either situationally—when someone is driving or cooking, for example—or systemically, as when someone is blind or isn't able to use their hands.

#### **INTUITIVENESS**

This is an area where conversational UI shines especially brightly. When effectively crafted, a conversation frees the user from having to know exactly how to do something before they try to do it. A well-designed conversational experience removes that burden with intuitive queries and natural language processing that captures the user's intent, and sophisticated



dialog design that helps users down the path of discovery until they learn what they want to learn or do what they intended to do. Imagine, for example, someone wants to connect their new wireless headset but isn't familiar with Bluetooth. Rather than navigating through the Settings menu, hunting for the appropriate setting, and trying a variety of things to see what works. In a well-designed conversational experience, all the person needs to do is say "connect my headset," and the dialog can walk the person through the tasks required, saving the person the mental headache of finding their way. Another example is troubleshooting a problem. "I can't print" can lead to a structured diagnosis flow that helps people solve their problem without even necessarily having to understand the nature of the problem.

#### **FMPATHY**

Conversations are inherently and inescapably emotional. Study after study has demonstrated that conversations authentically activate people's emotions, even when they know they're interacting with something artificial. In scenarios that deal with emotional circumstances, conversation is a powerful way to honor people's emotional reality. When a user feels helpless and frustrated—not to mention joyous or enthusiastic—it's as valuable to build kind and careful guidance as it is to offer accurate information. A thoughtfully designed conversation can create an opportunity for people to feel supported as well as be supported.

#### **TRUST**

Building a personality for your conversational experience is an effective way to create a connection with your users. Even if your conversational experience has no identity, no name, or no avatar, it still communicates through language, and language cannot help but communicate a persona. The quality of your dialog helps tell your user what to expect from your product or company.

An intentionally crafted persona communicates consistency, trustworthiness, and care. Even if your conversational experience is a sarcastic, belligerent crank with a cutting wit, if it's *consistent* with its personality, your users will know exactly what to expect and how to communicate effectively to achieve their goals. Personality makes it easier to teach your user how to speak to you.

Now add chit-chat and the palette for personality gets richer. Chit-chat represents the kind of casual conversational turns that don't necessarily relate to accomplishing tasks, but rather they allow your user to get to know the personality of your experience. A few carefully chosen chit-chat scenarios can go a very long way towards establishing rapport and loyalty, as well as smoothing over some of the inevitable misses in other conversational exchanges.

Thoughtfully crafting the personality of your conversational user experience will offer you some agency in how people respond to your experience and react with your brand.



## 03. Creating a responsible conversational experience

Because conversational design is still relatively new, the choices we make now will set precedent for future decisions, future models, and future habits. We're all learning as we go, and it's incredibly exciting to be exploring new possibilities with artificial intelligence, machine learning, and conversational experiences. Crucially, ethics are also critical for the success of your brand. Ethical responsibility and brand responsibility are tightly intertwined, but not necessarily the same thing. It's valuable to consider both when developing your conversational experience.

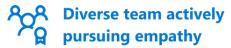
Designing a beautiful and effective conversation requires insight into what your user wants or needs to accomplish: completing a task, or getting unstuck, or having a curious conversation, or learning something they didn't know, to name only a few. But as we consider the design requirements that will make those experiences effective—the flows, greetings, errors, personas, and handoffs—it's equally critical to consider our other responsibilities to the people who use our designs. Will our choices exclude people, even inadvertently? Are our flows accessible? Are they transparent? Do they anticipate the possibility for misuse?

The design and authoring choices we make do have the potential to do great good, even in simple flows. Just as effective engineering design requires understanding the ways that a poor design can cause failure, a responsible design requires carefully considering the harm we might do if we're don't design thoughtfully and with intention. To do this, we recommend a few actions you can take:



#### ASSEMBLE YOUR TEAM

Design and engineering teams come in many shapes and sizes, from the one-person operation to the committed wing of a multi-national company. But small and large operations alike can create better and more responsible products when they can swap ideas, review their work, and bring in outside voices to help with questions of identity and accessibility.



If you do have the luxury to build a team, it can be enormously helpful to bring in people with a variety of lived experiences, as well as people who have a humanities background, or experience with creative expression. Novelists, musicians, playwrights, movie makers, poets... people who have experience writing from different points of view can help a conversational experience feel truly welcoming and effective.



#### CRAFT YOUR PRINCIPLES

If you were put on the spot and asked to articulate your values—that is, the things that matter most to you—how well could you do it? How well would your words and actions map to your core values?



These are weighty questions, but they are rewarding questions to weave into a design and development process. Even if your conversational experience has no identity, no name or avatar, no personality of any kind, the words and actions you choose—even what you choose not to say—will tell users what they can expect from your product. Can they expect kindness, transparency, tolerance, clarity, or candor? Do you allow flirting? Do you encourage it? Or do you discourage it? Can your agent reveal or discuss political ideology? How do you handle frustration or anger? What if that frustration turns toward abuse?

The process of creating design principles can help clarify for your whole team what you most want to represent in your experience, as well as identifying the things you want to avoid. Though our own biases may show in our advice, we're not trying to advocate for a particular moral framework for your project. You have the right to design your product to reflect whatever values you choose, even if others find them objectionable. However, we've observed that neglecting to think about the values you want to reflect in your conversational experience can expose you to recrimination, bad press, or brand exposure.

In fact, in some circumstances, you may also run the risk of embarrassing your project and brand by *not* having answers for some thorny questions. There are some common issues that arise in conversational exchanges that you'll want to consider, even if you don't author a single interaction that addresses them directly.

#### **Transparency**

Is it clear to users they're talking to an artificial voice rather than a human? If not, why not? Will people understand how to talk to a person if they want to?

#### Privacy and security

Can your conversational experience speak confidently to how you're handling personal or private data? Can you back up any claims made with concrete information on your website?

#### Chit-chat

How will you handle queries outside of your domain? Will you answer them at all? If not, what happens when someone strays off script?



#### Healthy boundaries

How will you handle abusive, violent, sexually explicit, or otherwise problematic queries?

#### Sensitivity

Does the language you use signal sensitivity to people's identities and abilities, including their mental health? Will you reply if someone reveals something sensitive about their state of mind or identity?

#### Clarity

Are you using jargon, technical language, or terms otherwise targeted at a particular audience? If so, is there a risk of alienating people unfamiliar with that language? Or is it appropriate for your audience?

#### Culture & nationality

Are you representing a particular cultural voice? If not, will people assume that your experience presents the dominant demographic of your region or country? And if so, are there any phrases you might avoid or embrace to avoid appearing culturally appropriative?

#### Identity

What identity markers are you signaling with your choices? Does your bot have a race? Gender? Sexual orientation? Age? If not, will people assume that it does based on the dominant demographics of your country or region? Are you comfortable with that assumption?

We've developed a <u>brainstorming worksheet</u> to help you consider questions like these. In our experience, it's useful to think these questions through, and use them to develop 3-5 design principles that best express the focus of your product. Too few and it's overly general, too many and the focus gets muddy. On our teams, we constantly refining these principles over time, and allow them to evolve as product needs change and our understanding deepens.

And to go back to the earlier point, the likelihood that your principles will reflect your best values goes up when you can review them with others. Even when you're a team of one, it's helpful to run them by people you trust to give you critical and honest feedback.

#### DESIGN WITH EMPATHY

Conversation is inherently emotional. When people converse, even about mundane topics, it triggers all sorts of emotional responses in one another. Those same emotional responses get churned up even when talking to a device. Even though users know they're communicating with an inanimate object, they still experience similar feelings as if they're talking to another human.



When building conversations, it's not only your responsibility to honor the emotional reality of the people using your products and services, it's also very much in the best interest of your brand. In addition to building a technically complete and effective conversational experience, it's also your job to think carefully about your user's emotional state and prepare conversations that meet them in that space as best you can.

It's certainly possible to fake empathy, or at least craft a convincing facsimile. Brands worldwide reckon with how to project sincerity without appearing craven or cynical, and it can indeed be a tricky line. A development process that's steeped in rigor about both goals and values will be more sincere and authentic.

#### **REVISE ONGOINGLY**

No matter how stellar your team and how airtight your principles, no matter how carefully you've considered accessibility, transparency, and issues of inclusivity, there is always more to learn, and our understanding will always continue to deepen and grow. We offer this same recommendation for making sure your design and engineering process is healthy: go back to make sure that what you've done is still working.

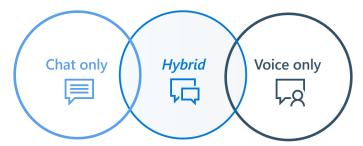
Although it's never too late to begin reviewing work that's already out in the world, it's easier to gather actionable insights if you build review options into your design up front. This includes using analytics options, especially for AI- or ML-driven work, but it also may include asking your customers for their insights.

We've included practical advice about how to do this in other topics, but when you think about building responsibly, a key element is to offer yourself the conditions to openly and courageously receive feedback that may challenge your assumptions. It is challenging to change direction, even when you're convinced it's important to do so, which makes it all the more important to practice that vulnerability and rigor, and to work with people who are equally committed to making sure your work truly represents your principles and meets the needs of the people who use your conversations.



## 04. Types of conversational experiences

Conversational experiences take on several form factors, and any given experience might start in one type of conversational user interface and continue in another, or the experience might be multi-modal. Some of the more common examples of the types of conversational experiences are included below with descriptions to help illustrate the key differences in the capabilities and functions that separate one bot from another.



It's important to note this list isn't exhaustive, and you may already be familiar with some of these terms or seen them used in a different capacity, as they are often used interchangeably. For example, a company may create a chatbot for customer support and refer to it as their support bot, while another company may build a bot with the same functionality and call it their virtual agent.

However, all of these terms refer to bots and share a common approach to many aspects of conversational experiences, even though the complexity, sophistication, and name of the implementation may differ.

#### **CHATBOTS**

Chatbots are one of the most familiar forms of conversational experiences that people have encountered. They are becoming ubiquitous fixtures on company websites, offering help with everything from buying a car to getting technical support. Typically, they are text-based. The user enters a question or request, and the bot responds. Depending on the user's stated intent, the bot might respond with a natural-language answer, or it might present something visual, like a map or a picture of a product. Or it might present an affordance for the user to take action, like adding an item to a shopping cart or picking a date and time for an appointment.

Chatbots give users a way to zero in on the information or action they need, without having to hunt around on a website. Bots make the search experience more intelligent by helping a user refine their search or disambiguate a general query.

Conversations with a bot can be question and answer pairs, where the bot doesn't maintain context from a previous exchange as the user asks another question. Or conversations can be multi-turn: more like human conversation, where each exchange builds on or follows logically from previous exchanges.



#### VIRTUAL AGENTS

Virtual agents are similar to chatbots, as they're built to help users complete a task. Their domain is usually limited to specific functions or skills, such as booking a flight, placing an order, or troubleshooting a technical problem. They might provide a conversational means for tasks like filling out a form or looking up information, allowing a person to focus on more complex activities.

#### VIRTUAL ASSISTANT OR PERSONAL DIGITAL ASSISTANT

Virtual assistants and personal digital assistants are usually branded personas, like Siri, Alexa, and Cortana. Their scope is not limited to a specific function or skill; they're able to complete any number of tasks on behalf of the user. They also present an extensible platform that new skill sets can be plugged into. Third party companies can build their own conversational experiences to allow the virtual assistant to be part of, say a ride-sharing service or a calendar reminder experience.

Virtual assistants are usually voice-activated, but they can converse in text modality as well. Because of their voice and their branded name, virtual assistants exude a deliberately crafted personality to create an emotional connection with the user.

#### **VOICE-ONLY INTERACTIONS**

As the name suggests, voice-only interactions are conversational experiences that rely on voice as the primary input. A graphical user interface might be completely absent. Examples include many factory-installed voice-activated navigation systems in cars, automated voice systems that provide customer service or support using the phone, and Bluetooth devices that provide voice-activated features without the use of a virtual assistant.

Because many of these experiences only use voice interaction, there's a need to anticipate potential variations in how users respond. For example, when asked for confirmation there are number of different ways someone may respond in the affirmative: "yes," "yup," "yeah," "uh huh." The varying degree of the volume, clarity, breaks and pronunciation of the words by the user adds to the complexity of voice-only interactions, and requires a high degree of adaptability to ensure the bot is able to accurately interpret, understand, and respond to user intents.

Furthermore, audio characteristics of the voice need to be taken into account as part of the design. Factors like timbre, pitch, and vocal inflection will convey human-like qualities to the experience, and designers will need to carefully consider implications like gender, culture, and personality that are conveyed by the voice. For more insights into implications of this kind, see <a href="Creating a responsible conversational experience">Creating a responsible conversational experience</a>.

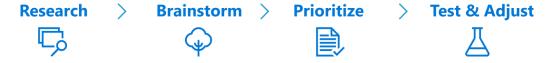
## **DESIGNING THE IDEAL EXPERIENCE**





## 05. Identifying key scenarios

Before you start creating dialogs or building content for your conversational experience, the first step is to scope your work and build a roadmap of key scenarios to guide you in the process. It's best to begin with a small set of content that will make the biggest impact with your users, and then gradually iterate and build upon that work. Here are four steps to help you get set up and running strategically.



#### RESEARCH & ANALYZE EXISTING DATA

Start by analyzing existing data and knowledge about how your customers are currently interacting with your brand. This might include website visits, blog traffic, social media engagement, search data, support calls, or any other source relevant to your users. Look for areas of high volume, high user engagement, or any other info that might be overall relevant to the purpose of the conversational experience. You want to identify all the potential areas that could be handled by your conversational experience.

#### **BRAINSTORM SCENARIOS**

After reviewing existing data, start brainstorming scenarios that you might cover with your conversational experience. Think carefully about your current user experiences, what the data tells you about those experiences, and list out topics that might help address your most frequently asked questions. You might start by answering these questions:

- Who is the intended audience for the bot?
- What are your business goals?
- What are your biggest problem areas?
- What are your users' biggest problem areas?
- What are the priorities for your business?

At this stage, don't worry about narrowing down your scenarios, prioritizing, or current resources and timeline. Simply make a list of all the topics you may want to cover with your conversational experience at some point. It can help at this point to dig into these scenarios and think through how users will interact with the bot in each use case.

#### PRIORITIZE FOR IMPACT

When you've finished brainstorming, the next step is to prioritize and choose a few scenarios to build in your conversational experience first. The key to this step is to start small in terms of number of scenarios, but big in terms of impact.



! Note: Don't throw away your brainstorming once you're done! Those ideas will be the building blocks for future content development, and they will act as a basis for your innovation roadmap.

When narrowing down your initial scenarios, take into account the data you analyzed and your answers to the questions above about business goals, problem areas, and priorities.

#### TEST, MONITOR, & ITERATE

The process of scenario development should never be stagnant. Even after you've begun developing content for your conversational experience, continue to monitor usage and analyze data to make sure it addresses the most pressing, important, or impactful user scenarios. Continue to update your content plan over time as needs change.

#### **EXAMPLES OF SCENARIO DEVELOPMENT**

Here are two examples of how these steps can be put into action to develop scenarios for content experiences. In each example, available data is analyzed, possible scenarios are brainstormed, and then a list of prioritized scenarios is created based on balancing the business goals with the scenarios that will have the biggest impact on users.

Example 1: Customer support bot for a mobile phone retailer

Available data	<ul> <li>Questions asked on social media channels</li> <li>Traffic data from the frequently asked questions (FAQ) page on website</li> <li>Call volume to customer support</li> </ul>				
Brainstormed scenarios	Questions about mobile phone features, pricing questions, troubleshooting mobile data issues, troubleshooting automatic factory reset error, ordering replacement parts, repair services, wireless carrier compatibility, billing issues, new phone release information, shipping information, service outages, warranty information				
Prioritized scenarios	<ol> <li>Troubleshooting mobile data issues</li> <li>Troubleshooting automatic factory reset error</li> <li>Repair services</li> </ol>				
Why these scenarios were prioritized	<ul> <li>The first and second scenarios generate the most customer support call volume.</li> <li>The third scenario has the highest traffic on the FAQ website.</li> </ul>				



Example 2: Personal shopping assistant for clothing brand

Available data	<ul> <li>Past sales data</li> <li>Current industry trends</li> <li>Traffic data from the frequently asked questions page on website</li> </ul>				
Brainstormed	Clothing recommendations, accessory suggestions,				
scenarios	completing purchases, returns and exchanges, size and fit				
	information, current discounts and sales, shipping				
	information, what's new, store locations, store hours,				
	inventory information				
Prioritized	1. Clothing recommendations				
scenarios	2. Size and fit information				
	3. Returns and exchanges				
Why these	The first scenario is the main reason for creating the				
scenarios were	shopping assistant.				
prioritized	The second and third scenarios have the highest traffic on the FAQ website.				



## 06. Understanding and identifying customer intents

After prioritizing the <u>key scenarios</u> you want to focus on, the next step is identifying the user intents you'd like to address within each scenario.

#### WHAT IS A USER INTENT?

You can think of user intent as the reason why a person is interacting with your conversational experience. Whether the user asks for something specific or expresses themselves in a way that requires clarification, you'll need to identify what the user is actually requesting. Identifying the true intent ensures your users are matched with the most accurate content to help them complete their goal.

User query:

"The jeans I bought last week are too big. Can I get them one size smaller?"

Intent: exchange an item for a different size

#### WHY IS IT IMPORTANT TO IDENTIFY USER INTENTS?

Although this may seem fairly straightforward, it's important to take the time to thoroughly explore the different possible underlying reasons that would bring a person to your bot. This will provide valuable insight into your user's actual intent. If the user isn't matched with the correct intent, they may receive wrong or misleading information, causing frustration and the potential loss of trust in your brand.

#### **GETTING STARTED WITH USER INTENTS**

A good place to start when identifying intents is by reviewing your key scenarios and any available data that show how users are currently interacting with your brand or conversational experience. Start with a single scenario and think critically about what your user's purpose might be in that scenario. Keep in mind there may be multiple intents per scenario.

Here's an example of potential user intents for a given scenario of a conversational experience as it walks through the development of user intents.



#### Example: Personal shopping assistant for clothing brand

Conversational experience	Personal shopping assistant for clothing brand				
Scenario	Returns and exchanges				
User intents	<ul> <li>Return an item</li> <li>Return a defective or damaged item</li> <li>Return an item after 90-day return policy</li> <li>Exchange an item for a different size</li> <li>Exchange an item for a different product</li> </ul>				

Don't worry if you feel like you aren't capturing all customer intents. In fact, it's better to start with a smaller set of intents and make sure the ones you identified ladder back up to the goals and priorities you defined at the beginning. Just as you would when defining scenarios, it's best to keep the scope small when you are beginning to develop content will help focus your efforts on the most important and high-impact areas.

#### INTENTS AND QUERIES

It can be easy to confuse intents with queries. Intents define the goal of the interaction, while queries are or the context-setting words, phrases, or questions that users say or type to begin a conversation. It's important to note that you cannot rely on a single query to accurately describe an intent. You may discover similar queries are actually multiple intents that require separate content, or you may find there are different ways to ask about a single intent. To learn more about how to intents and queries are related, see <a href="Queries">Queries</a>, utterances, and triggers.

#### MONITOR AND ITERATE YOUR INTENTS OVER TIME

Taking the time to properly understand and identify intents ensures the content you create will help users as much as possible. Identifying user intent, much like any content you create for a conversational experience, requires you to identify, implement, and monitor the success of the content. Let the data you collect guide how you'll refine intents, and make sure to map intents back to real user needs with matching queries.



## 07. Disambiguating customer intent

Disambiguation is the process of narrowing down a user intent by asking clarifying questions, or in some cases presenting options to choose from, to get a better understanding of the true intent and direct to the correct content. Disambiguation is helpful when you have multiple intents that are similar or when the user query is unclear. Whether disambiguation is necessary depends on the type of scenarios you support, how you identify and route different intents, and the content you have available for users.

For example, the query "How do I upgrade my computer?" may seem straightforward, but it requires more information to ensure the user's intent is matched with the most accurate content to complete the task:

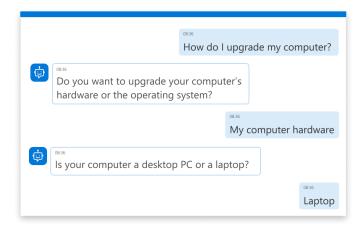
- Do they want to upgrade their operating system or the computer hardware it runs on?
- Is there a different process for upgrading for desktop PCs versus laptops or tablets?
- Do they want to upgrade their operating system to the newest version?
- Do they want to update their current operating system with security patches?

#### DIFFERENT APPROACHES OF DISAMBIGUATION

There are a few ways to approach disambiguation, including asking more questions, providing options, or asking targeted questions.

#### Ask more questions

The first way to disambiguate is to keep asking questions that help you narrow down the user intent. Using the same example as above, the conversation may look like this:

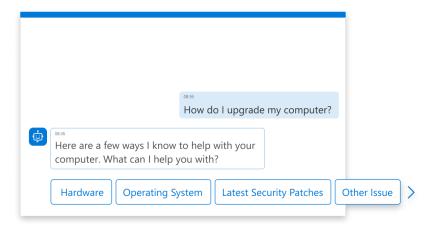


This requires continuing a line of questioning until the user's intent is identified and matched with your content. This method will lead to more conversational turns before an answer is provided. Clarity is important, but if it unnecessarily extends an interaction or makes an interaction feel more like an interrogation, it may lead to user frustration.

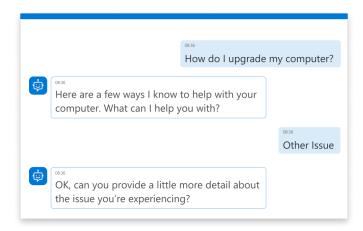


#### *Provide options*

Another approach to disambiguation is to provide suggestions up front to help users disambiguate more quickly and reduce the need to ask multiple questions.



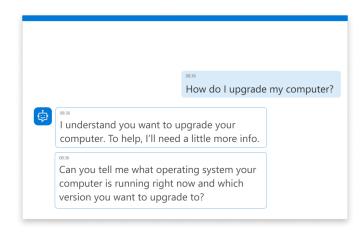
This method is upfront about what kind of information is available to the user, but if none of the options available address the user intent, there needs to be a way for the user to express that and be redirected to the correct information. This can be done by including options such as "None of these," or "I have a different question." If the options shown don't address the user's needs, you might then follow up with how the user might provide more context or details.



#### Ask targeted questions

The third approach to disambiguation is to use targeted questions that inform the user of what information is necessary to fully understand their intent. This approach is different than the first method because it can build upon the context understood from the initial query to ask specific questions and get the user to the best possible content faster.





Similar to the other options, the goal is to try to understand the user's intent as quickly as possible, without accidentally sending them to irrelevant content. However, these types of open ended questions have the potential to introduce more complexity into the interaction if the user includes too much detail in their response that causes confusion with the bot or ends up triggering incorrect content.

#### PREPARING FOR OUT-OF-SCOPE QUERIES

You may find some cases where disambiguation is impossible because the user intent falls outside of the scope of work for your bot and you don't have the content to support the scenario. It's okay not to have an answer for every question a user may ask, but, if possible, try to identify out-of-scope queries that are likely to arise and prepare answers that make it clear you understand and, if possible, help people find the assistance they need. This will prevent people from reaching a dead end, and can also provide valuable data if you see an increase in traffic to areas you originally planned to be out of scope.

For more information on how to prepare for the unexpected, see <u>Graceful fallbacks and handoffs.</u>

## PRACTICAL TIPS





## 08. Queries, utterances, and triggers

Though they have many names—queries, questions, triggers, phrases, utterances—these are the context-setting words provided by the user to begin a conversation. Often, they will be presented at the beginning of a conversational experience, but they may appear any time that a user's intent changes. We'll refer to them as queries in our guidance, but you may see a different term used to describe the same thing depending on the product or services you're using to create conversational experiences. Queries can be a single word, a group of words, a full sentence, a question, or even a paragraph.

#### WHY ARE QUERIES IMPORTANT?

It's the responsibility of your conversational experience to correctly identify and understand the user's query in a variety of forms, match it to an intent you've created, and present the correct response in return. Your ability to fulfill this responsibility may depend on the sophistication of the language understanding capabilities of the tools you use to create the conversational experience.

#### CREATE UNIQUE QUERIES

As you build a conversational experience, develop a library of queries that are unique to one of your identified user intents. This allows you to capture the correct conversation when users present any variety of queries pertaining to that specific intent. In most cases, it's a good idea to start with a limited number of distinct queries per intent (three to five queries) and add or remove queries over time as users interact with your experience.

#### BRAINSTORMING QUERY IDEAS

Here are some questions to help get you started with considering your queries.

#### *Try it out loud*

How might a user phrase their questions or statement if you were speaking with them in-person or over the phone? It can help to roleplay this scenario to identify more natural queries.

#### Brainstorm vocabulary

What kind of words would a user choose? Are they different from the industry terms? For example, will someone look for "inexpensive flights" or "cheap tickets"? Make sure to identify any common slang, contractions, acronyms, synonyms, or other variations for terms a user may use.



#### **Build variations**

Take into account that the same intent may have a variety of queries in many forms, such as a collection of keywords, a question, or a statement.

#### Keep track

Monitor your user data and make sure to refine and update queries as necessary over time, and to remove any phrases that are causing confusion.

#### RELATIONSHIP BETWEEN QUERIES AND USER INTENTS

As a rule of thumb, you should identify the user intent first, and develop queries second. For more information on how to identify user intents, see <u>Understanding and identifying user intents</u>. Here are a few things to keep in mind as you begin compiling queries for a user intent:

#### Consider different queries to express a single intent

There are many different ways someone can express their intent. For example, consider these two queries:

Query 1: When is your birthday?

Query 2: What day were you born?

In both, the intent is the same: to get information about your date of birth. However, the questions are asked in two different ways using completely different words. Think about the different ways users might phrase their questions, or choose to use different words that communicate the same intent.

#### Anticipate similarly worded queries with completely different intents

There are some cases where a query could potentially have more than one intent. For example, if a user recently made a purchase and asks:

Query: How do I return my purchase?

You might respond with details on how they can return an item and get a refund. However, it's possible the user actually meant they want to exchange their purchase for a different item or they want to know if they can ship it to a store location or if they need to return the purchase inperson. If you support multiple intents that are similar or related to one another in some way, you'll need to make sure they're clearly differentiated. Ensuring that your natural language model is well trained and can distinguish different intents is critical. Using disambiguation questions can also help narrow down the meaning of the user's true intent.



#### Identify critical differentiation words

When identifying the key words or phrases used for a query, think about which words are important to help set context and differentiate between intents. Think of the difference in the meaning for these two queries:

Query 1: I want to book a flight to Shanghai.

Query 2: I want to book a flight from Shanghai.

By changing a single word, the intent behind those queries has changed and would require two different responses. Make sure to think about how to handle these types of subtle differences in the queries you're creating for each intent and author appropriately.

#### **AMBIGUOUS QUERIES**

In some cases, a query may not be specific enough to confidently match a predefined intent. For example, someone might simply type "I need help," or "I have a problem with a product." In these scenarios, you can gather additional context with the use of disambiguation questions that help guide the user in refining their intent. For more information on how to do this, see <a href="Disambiguating custome intent">Disambiguating custome intent</a>.

#### **EXAMPLES OF POTENTIAL QUERIES**

Here are two examples of developing potential queries for two separate user intents that are within the same scenario for troubleshooting laptop display issues.

Example 1: Laptop display is only in black and white

Scenario	Troubleshooting laptop display issues				
User intent	Display color troubleshooting				
Potential queries	Screen color is missing				
	<ul> <li>Display is in black and white</li> </ul>				
	<ul> <li>Display isn't showing color</li> </ul>				
	<ul> <li>Change screen to show color</li> </ul>				
	<ul> <li>Screen stuck in grayscale</li> </ul>				
	<ul> <li>Color settings for screen</li> </ul>				



## Example 2: Laptop won't connect to external display

Scenario	Troubleshooting laptop display issues			
User intent	Monitor connection troubleshooting			
Potential queries	<ul> <li>Monitor connection troubleshooting</li> <li>Can't extend laptop screen</li> <li>Troubleshoot external display issues</li> <li>Can't connect laptop to external display</li> <li>External display doesn't work wit laptop</li> </ul>			



## 09. Graceful fallbacks and hand-offs

For situations where you can't adequately determine your user's intent, or when you simply can't fulfill the intent, you'll need to develop a series of fallback responses. We say "series" intentionally here; thinking of fallbacks as an isolated "Oops, sorry about that," won't be enough. When crafting fallbacks, it's important that you don't seed undue frustration or break the user's trust by creating an experience that is repetitive or stalls the user's progress. Your goal is to help the conversation get back on track as quickly as possible so the user can complete the task they set out to accomplish.

#### SET CLEAR EXPECTATIONS UP FRONT

Bots aren't appropriate for handling some tasks; some things are simply better handled by a human. During your design process, you identified the things your bot will do and the things it won't. One way to reduce the need for fallbacks is being clear right from the start what those things are.

For example, if you're designing a banking bot, tell customers in the greeting that they can check their balance or make a transfer between accounts. If you're designing a travel bot, tell customers that they can book a round-trip flight, reserve a hotel room, or make changes to their itinerary. Setting these expectations at the beginning of the interaction helps guide customers in the right direction and may even reduce the need for fallbacks.

When you get to the point where a user has asked your bot to do something that it can't, a fallback response is an additional place that you can provide this clarity. Fallbacks like, "Sorry, I didn't get that. I can help you <X> or <Y>. Do you want to try one of those things?" can help redirect the user toward the things your bot can do.

#### THINK OF FALLBACKS ACCORDING TO FUNCTION

Regardless of whether your bot can't understand or is unable to deliver on what user the wants, it's helpful to think of your fallbacks according to their function:

#### Seeking understanding

This is your chance to ask the user to describe or rephrase their request in a way that makes it easier for your bot to understand their intent.

#### Examples:

- I didn't get that. Can you say it another way?
- I don't quite understand. Can you try rephrasing?
- I'm a little unsure how to help. Try asking again using just a few key words.



#### Finding additional ways to disambiguate

Sometimes, a fallback is an appropriate place to seek additional clarity that will help you further determine what the user wants. Offering a suggestion or two that closely match the user's intent can help.

#### Examples:

- Did you mean [suggestion]?
- Sounds like you want to [suggestion]. Is that right?
- I found [suggestion 1] or [suggestion 2]. Is it one of those?

#### ESTABLISH DOMAIN EXPERTISE

If your bot understands the intent, but is unable to fulfill it, be transparent with your users. Redirect them toward what the bot can do or offer additional resources that could help.

#### Examples:

- Sorry, I can't help with that. Did you want to try [suggestion 1] or [suggestion 2]?
- Sorry, I don't think I can help you with that. Say "main menu" to learn what I can do.
- I don't have any info about that, but I found this topic that might help: [topic].

Be mindful about using phrases that suggest the bot will learn how to address the user intent like "I can't do that yet," or "I'm still learning how to do that," unless you have concrete plans to build that capability into your bot.

#### CREATE FALLBACK VARIATIONS

If you were having a conversation with someone, and they made several errors in a row, it would be odd for them to offer the exact same apology over and over again. The same goes for bots. If all your bot can say is, "Oops, sorry about that," the repetitive nature of the response will quickly become irritating. When writing fallbacks, it's a good idea to create a few for every situation. That way, when customers encounter a fallback more than once, the experience won't feel overly robotic. The number of fallbacks you'll need depends on how many paths customers can follow in your conversation flow, but generally, try to write at least three.

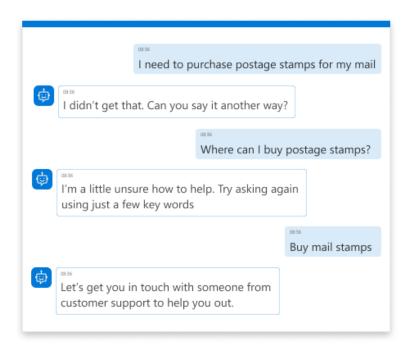
#### KNOW WHEN TO HAND OFF

Similar to variations in fallback responses, it's important to build a hand-off process for when your bot just isn't understanding the user or isn't able to help them. Your bot might direct the customer to a human support agent so the user can get further assistance or to available resources such as support websites or online documentation. One of the trickier questions you'll encounter is: When is the right time for the bot to direct the user to a human or other resource?

It can be helpful to think about how many times you would be willing to repeat or rephrase a question to someone during a conversation before becoming frustrated. Our recommendation



is ask the user **no more than two fallback questions** in one session before directing them elsewhere. If your bot isn't able to understand the user's intent after they try to communicate it three times (including their first question), chances are, a fourth time restating it will not help the bot understand.



Try making the hand-off as smooth as possible. Make sure the user knows what is happening, whether they'll be connected to a human or another resource, and what they'll need to do next. Remember that even if they're stuck, that doesn't necessarily mean they need to start over—and they won't want to. A good hand-off will effectively remember where the user left off and help them continue the task they set out to accomplish. Asking the user to repeat the same process the bot initially started isn't a great experience and may result in the user abandoning the process.

#### ASK FOR FEEDBACK

When your bot concludes the conversation, whether it's been able to help the customer or not, it's a great time to ask for feedback. Make it simple and not at all time-consuming. Here are some easy ways to ask people about their experience:

- Thumbs up/thumbs down
- Smile/frown
- Numeric rating (five-point scales are typical)
- Positive/negative (either a binary scale or a broader five-point scale)

Ideally, include an open text field below the rating so the customer can say whatever they want. You can add more questions, but the more questions you add, the less likely people will be to engage with the feedback form.



As valuable as feedback can be, though, it's equally important to be thoughtful about how often you ask for it. Asking too often will be annoying at best and alienating at worst. If possible, try to use frequency signals from your customers so that you're not asking them for a rating more than once a week. Even then, you may want to prioritize experiences where feedback would be most useful, such as new experiences or more complex ones. You may also want to avoid asking for feedback after experiences that may be more urgent or where the user may want to quickly move on to something else, such as getting a phone number. In other words, try to make sure they task they set out to complete is done before distracting them with the task of completing your survey.

Big caveat: do not ask for feedback if you have no way to manage it. If customer feedback goes into a void, and there's no process set up for who will review, label, tag, store, and report out on it, don't bother.



## 10. Error handling

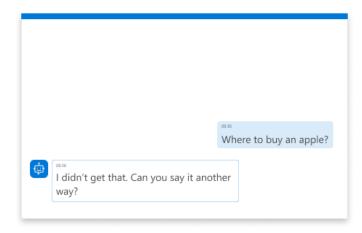
What happens when your bot gets something wrong? What ways do you offer users to correct the error? Error handling is a form of fallback and can apply to user interactions with the bot, as well as to the functions of the bot. For more information on fallbacks and hand-offs, see <u>Graceful fallbacks</u> and hand-offs.

Different types of errors may require different messaging, but regardless of the cause, try to be as transparent as possible to help the user. This may include details of the error itself, but most importantly, include info on what the user can do to get unstuck—or at least understand why they're stuck.

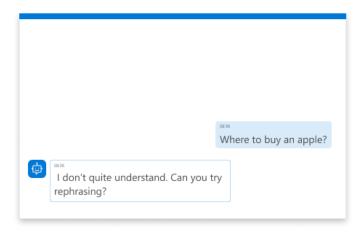
#### MISRECOGNITION ERRORS

In a situation where your bot doesn't understand the user's intent, there are a few different ways to ask for clarification:

Example 1: Ask the customer to rephrase.

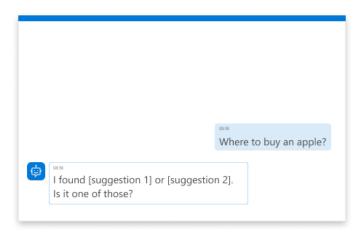


Example 2: Ask the customer to repeat themselves.





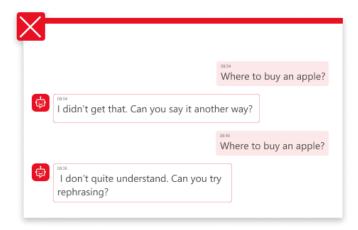
Example 3: Present the customer with specific options.



As language understanding improves, this will become less of an issue. For now, asking for clarification can help mitigate user frustration and get your customer to the right place in the conversation flow.

If the misunderstanding happens more than once, choose your responses carefully. If in the first response you ask the user to repeat themselves or rephrase, offer specific options in the second response. We don't recommend having the user repeat themselves more than twice. If your bot can't understand the user's intent in two turns (that is, after two queries and answers), hand off to a human or other issue-mitigation tactic, like email.

Example: Do not ask the user to repeat themselves more than twice.

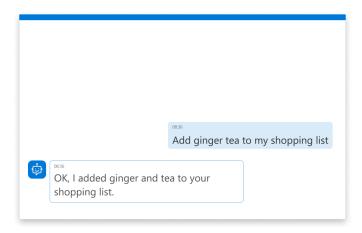


#### **EXECUTION ERROR**

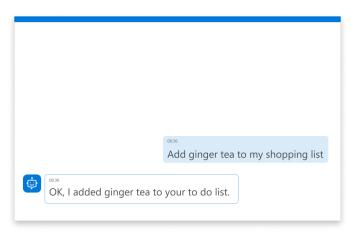
Execution errors are ultimately the result of misrecognition errors: the user has asked for something, and your bot has either come up with an incorrect match for the request or misunderstood the input the user has given.



Example 1: Misunderstood input (two items added to list instead of one)



Example 2: Incorrect match for user's request (item added to wrong list)



When this happens, try to provide a way for the user to go back and fix the bot's error.

- Recognize secondary inputs like "wrong list," "that's not right," or simply "no."

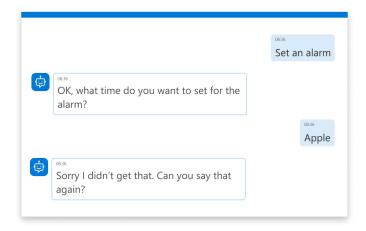
  Users have an automatic impulse to immediately fix any errors, and you can use that to your advantage. From a technical standpoint, implementing this type of understanding can be challenging, especially if the interaction model is a question-and-answer design instead of a multi-turn conversation. However, taking the time to integrate these types of keyword phrases into your design for error handling can help make the user's experience much more natural.
- Ask for confirmation before executing an action. If the interactions above had
  included a confirmation like, "OK, I'll add ginger and tea to your grocery list. Is that
  correct?" the user would then have the opportunity to correct the error before it
  happens. However, asking for a confirmation with every small action can feel tedious,
  repetitive, and ultimately annoying, so use this approach with caution.



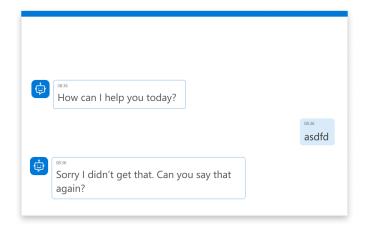
#### INPUT ERROR

Sometimes, your bot isn't necessarily to blame for the error. It's possible the user has said something that isn't an answer to the question the bot asked, or that they've said something that is unintelligible to the bot. In these cases, a simple, "Sorry, I didn't get that. Can you say that again?" is acceptable.

Example 1: Unexpected user response to bot



Example 2: Unintelligible input



#### **TECHNICAL ERRORS**

Another set of scenarios to take into consideration are errors that cause the bot to not function correctly. An example of this is when there's an issue with the user's internet connection. Types of technical errors may include:

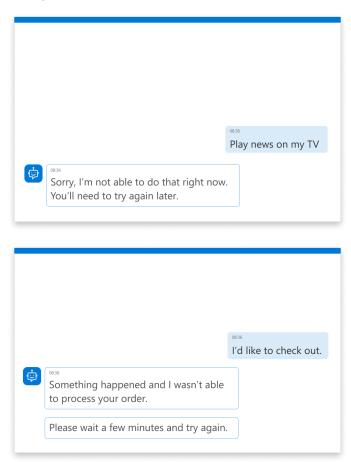
- Errors with a solution. You're aware there's something wrong with how the bot works
  and have a workaround or fix available for the user to try, or you're actively working to
  fix the error.
- **Errors without a solution.** You're aware there's something wrong with how the bot works, but there currently isn't a workaround or fix available.



• **Unknown errors.** Something happened and the bot doesn't work correctly. Essentially a generic error, for when the cause of the error isn't clear.

If you include responses that handle technical errors, make sure to let users know there's an issue with the bot.

#### Examples:



It's helpful to let the user know when they should try again, but only include that info if they'll have a higher likelihood of success when they try again. If they try again and see the same message, it can negatively impact their trust in your bot.

If your error is either unknown or doesn't have a solution, be sure to direct the user somewhere else, like to a customer support agent, FAQ, or email or chat support.



## 11. Ending conversations

When building a conversational experience, it's important to consider when and how your bot will end a conversation with someone. Working through this process will help you create a much more realistic experience, as real conversations have endings. But how do you know when a conversation has ended?

Think about your own conversations with other people, whether it's in-person, online, or over the phone. There are number of different signals that may indicate a conversation is about to end: a lull in the conversation, an awkward transition to an abrupt goodbye, and even subtle signals like body language or how engaged the person appears to be.

Chat bots aren't necessarily receptive to those human subtleties, though. Due to their reactive nature, conversational experiences with bots work best when the bot can help guide the conversation. The most accurate way to know if the bot has reached the end of a conversation is to ask the user if they're done.

#### ASK A FOLLOW-UP QUESTION

After completing a user's request, asking a simple follow-up question will help the bot understand if it was successful.

#### Examples:

- "Did this answer your question?"
- "Did that fix the issue you were experiencing?"
- "Can I help with anything else?"
- "Do you need help with anything else?"

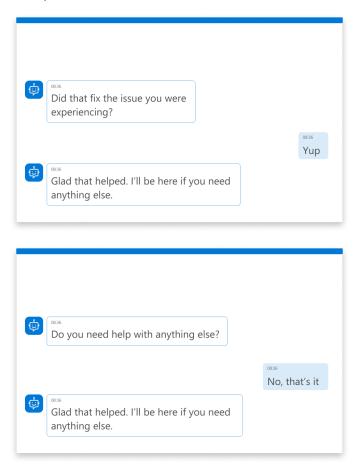
Checking if the user is satisfied with the assistance or information they received is a way to confirm they were able to complete what they intended to do when they started the conversation. If the user responds they didn't get the help they need or they have additional requests, it signals the conversation isn't over and should continue.

#### DON'T JUST SAY GOODBYE

When the user confirms a conversation is ready to end, using just "Goodbye" or "Bye!" may come off as abrupt or make it seem like the user doesn't have the opportunity to continue with a different conversation. A better approach is to let the user know the current conversation has come to a close, and leave open the possibility for further interactions.



#### **Examples:**



In both of these examples, when the user responds they are ready to end the conversation, the bot reconfirms the user's response and lets them know they can always ask another question later.

# TOOLS

## A BRAINSTORMING WORKSHEET

IDENTITY								
NAME IDEAS (Does it have a name? Or just a title?)			<b>REGIONS</b> (Where will the bot be used?)					
ICON IDEAS (Brand icon, glyph, full avatar, etc.)		LANGUAGES (Will you support others besides your own?)						
AUDIENCE								
WHO USES YOUR BOT			3 REASONS THEY USE IT What do people use your bot for? What are the top things it helps people do?					
Who is your audience? Describe them here.		REASON 1		beopie do:				
			REASON 2					
		REASON 3						
		tics of your brand—and therefore, the	key characteristics of your bot					
<b>5 TRAITS WE EMBODY</b> What are your brand's top 5 characteristics? Write them down here.		SIMILAR CHARACTERS Identify actual people or characters who talk like your brand.		<b>5 TRAITS WE AVOID</b> What are the top 5 characteristics your brand avoids? Write them down here.				
		BRAND EXAMPLES  Are there any brands that have a voice similar to yours?		_				
TRAITS This is how your voice changes depending on 3 REASONS PEOPLE USE MY BOT		the situation.  3 THINGS THAT COULD HAPPEN		HOW MY BOT RESPO	INDS			
Now, take the things you wrote in the <u>gray box</u> and write them again down below.		For each reason someone might use your bot on the left, write down a <u>specific situation</u> one of your customers might find themselves in.		Rewrite each of the 5 traits you ca Then, circle on the scale of 1 to 5 to invokes this trait in the situation a	me up with in the <u>blue box</u> . he extent to which your bot			
REASON 1				Trait 1:	12345			
				Trait 2:	1 2 3 4 5			
				Trait 3:				
				Trait 4:	-			
				Trait 5:	-			
REASON 2				Trait 1:	12345			
				Trait 2:	1 2 3 4 5			
				Trait 3:				
				Trait 4:				
				Trait 5:				
REASON 3								
NEASON 3				Trait 1:	12345			
				Trait 2:	1 2 3 4 5			
				Trait 3:	1 2 3 4 5			
				Trait 4:	1 2 3 4 5			
				Trait 5:	1 2 3 4 5			
CONTENT STYL	This is how your tone is exn	ressed. Or: how your bot talks, like the	e words it says and how it says	them				
QUICK LIST	PUNCTUATION	ressea. Or. now your bot turks, like the	. Words it says and now it says	them.				
Emojis								
GIFs & memes	SENTENCE CONS	SENTENCE CONSTRUCTION						
Images	CAPITALIZATION							
Uideos  JARGON & SLANG								



## TRY WRITING AN ACTUAL CONVERSATION HERE. OR, JUST JOT DOWN MORE IDEAS.



# This CUX Guide was a collaboration between conversational teams at Microsoft

## **Content Intelligence Team**

Deborah Harrison, Mary Dusek, Stephanie Blucker, Tatsuo Yamada, Roxanne Kenison, Allison Heisel and Kat Noel.

#### **Conversational AI Team**

Quirine van Walt Meijer, Xi Yang, Em Ivers and Marieke Iwema.

## **Get started with Azure Bot Services**

