

CHAPTER 1

The Beloved Chatbots

When you begin to build a chatbot, it's very important to understand what chatbots do and what they look like.

You must have heard of Siri, IBM Watson, Google Allo, etc. The basic problem that these bots try to solve is to become an intermediary and help users become more productive. They do this by allowing the user to worry less about how information will be retrieved and about the input format that may be needed to attain specific data. Bots tend to become more and more intelligent as they handle user data input and gain more insights from it. Chatbots are successful because they give you exactly what you want.

Does it irritate you or frustrate you when you have to enter the same name, e-mail ID, address, and pincode every time on different websites? Imagine a single bot that does your tasks—say, ordering food from different vendors, shopping online from various e-commerce companies, or booking a flight or train tickets—and you don't have to provide the same e-mail ID, shipping address, or payment information every time. The bot has the capability to know this information already and is intelligent enough to retrieve what is needed when you ask it in your own language or in what is known in computer science as Natural Language.

Chatbots development is way easier than it was a few years ago, but chatbots did exist decades ago as well; however, the popularity of chatbots has increased exponentially in last few years.

If you are a technical person or have some idea of how a web application or mobile application works, then you must have heard the term APIs. Any kind of data that you need today is available to be consumed in the form of APIs provided by different service providers and institutions. If you are looking for weather information, booking tickets, ordering food, getting flight information, converting one language to another, or posting on Facebook or Twitter, all of this can be done using APIs. These APIs are used by web- or mobile-based applications to do these tasks. Chatbots can also use these APIs to achieve the same tasks based on our requests.

The reason Chatbots get an edge over traditional methods of getting things done online is you can do multiple things with the help of a chatbot. It's not just a chatbot, it's like your virtual personal assistant. You can think of being able to book a hotel room on booking.com as well as booking a table in a nearby restaurant of the hotel, but you can do that using your chatbot. Chatbots fulfill the need of being multipurpose and hence save a lot of time and money.

In this book we are going to learn how to build natural conversational experiences using bots and how to teach a bot to understand our natural language and make it do tasks for us from a single interface.

Bots in general are nothing but a machine that is intelligent enough to understand your request and then formulate your request in such a way that is understandable by other software systems to request the data you need.

Popularity of Chatbots Usage

Chatbots have become popular just as anything from the recent past. Let's try looking at Figure 1-1, which depicts the rise of chatbots, and also try to understand why there is a huge demand for building chatbots.



Figure 1-1. Numbers on Y-axis represent search interest relative to the highest point on the chart across all categories worldwide

The simple answer that comes to mind is that it's not a complex software and can be used by anyone. When we build software we target the audience who will be using it, but when it's used by anyone else, it becomes difficult and unusable. When we build chatbots we keep in mind that it will be used by people of all age groups. This happens only in case of chatbots, where the software tries to behave like a dumb person (but it's intelligent) and lets the user be who he or she is. In all other software, you will find that you should be aware of some terminologies or gradually be aware of how to optimally make use of it, but that's not the case with chatbots. If you know how to talk to a person, you won't have any issue using a chatbot.

There is a continuous growing demand for chatbots. However, there has not been much research that has empirically tried finding out the motivations behind using chatbots. In a recent study, an online questionnaire asked chatbot users ages 16 to 55 years from the US to describe their need for using chatbots in their daily lives. The survey revealed the "productivity" to be the primary motivational factor for using chatbots.

The Zen of Python and Why It Applies to Chatbots?

I remember the Zen of Python, which says, "Simple is better than Complex," and that applies so many places in software.

The Zen of Python is a collection of 20 software principles that influences the design of Python Programming Language.

—Tim Peters

Want to know "What is Zen of Python?" Try the below steps.

If you already have Python installed on your computer. Just go to your Python interpreter and import this:

```
Python 2.7.15 (default, May 1 2018, 16:44:08)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
```

CHAPTER 1 THE BELOVED CHATBOTS

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one—and preferably only one—obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea—let's do more of those!

You may not be able to make sense of all points above relating to chatbots but surely you can most of them.

Well, coming back to our topic, I remember finding difficulty starting to use Facebook User Interface while coming from Orkut background. If you have never used Orkut, you would not understand it, but just try thinking of a situation in your life where you started using some new software or application and you had a hard time getting the hang of it. Maybe switching from Windows to MacOS/Linux or vice versa? When you use a new application, you need to learn a few things, and it takes time to get used to it and to know what it does and how it works. It does happen at times that you come to know some features of the application even after years of using it. If you are on MacOS, try Shift + Option + Volume Up/Down and see what happens. Let me know if it amazed you, if you didn't know it already.

In the case of chatbots, the communication between the user and the server or backend system is pretty simple. It's just like talking to some other person using a messaging app.

You just type what you want, and the bot should be able to either give you what you want or should guide you how to get that. In other words, it should point you to the correct information by giving you a link or document. The time has come where bots are able to even dig up the information from an article and document and provide it to the users.

Significant progress in AI by companies like Google, Facebook, and IBM and by machine learning services like Amazon Lex, wit.ai, api.ai, luis.ai, IBM Watson, Amazon Echo, etc. has led to the extraordinary growth and demand of such robots.

The Need for Chatbots

Now, we will try to look at the need and demand of chatbots in this fast-growing information creation and retrieval age from two different perspectives: the business standpoint and the developer's perspective. So, if you are a product manager, sales manager, or from marketing or any related domain that drives the business directly, then you should not skip the business perspective of the chatbots. It will give you a clear picture that businesses today need to adopt this technology to drive more revenue.

The Business Perspective

We will try to look at the business perspective of the chatbots. Is it good for a business to have a chatbot or to migrate lots of stuff to be done by chatbots?

The time has already come for businesses to treat chatbots as one of the marketing tools of this generation.

- **Accessibility:** They are easily accessible. The consumer can open the website and start asking questions or begin resolving their queries without having to dial a number and follow the ugly way of “Press 1 for this and Press 2 for that” in the IVR. They can quickly get to the point with just a basic set of information.
- **Efficiency:** Customers can sit at their desk in their office or on a couch in their living room while watching a game and get their status of a credit card application, find their food order status, or raise a complaint about any issue.

If you make customers efficient and productive, they start loving you. Bots do exactly that and help boost business.

- **Availability:** Chatbots are available 24 hours per day, 7 days per week. They would never ask you for leaves or get tired like human employees. They will do the same tasks or new tasks every time with the same efficiency and performance. You must get frustrated when some customer care phone number says, “Please call us between 9:00 AM and 6:00 PM,” just for a piece of information. Your bots would never say this.
- **Scalability:** One Bot => 1 million employees. You see this? Yes, if your bot can do what a customer needs, it can easily handle hundreds of thousands of customer queries at the same time without breaking a sweat. You don’t need to keep your customers waiting in queue until the customer representative becomes free.
- **Cost:** Needless to say it saves a hell of a lot of cost for the business. Who doesn’t like to save money? When bots do that for you, there is no reason why you shouldn’t like them.
- **Insights:** Your sales representative might not be able to remember the behavior of the user and give you exclusive insight about the consumer behavioral pattern, but your bots can using latest techniques of machine learning and data science.

Chatbots Bring Revenue

Chatbots have proven to be successful in bringing more revenue to the business. Businesses starting with chatbot support or creating a new chatbot to support customer queries are doing well in the market compared to their competitors.

As per one of the blogposts on stanfy.com, in the first 2 months after introducing its Facebook chatbot, 1-800-Flowers.com reported that more than 70 percent of its Messenger orders were from new customers. These new customers were also generally younger than the company’s typical shopper, as they were already familiar with the Facebook Messenger app. This significantly increased their annual revenue.

One of the greatest added values of chatbots is using them for generating prospects. You can reach your potential clients directly where their attention is (messengers) and present them your newest products, services or goods. When a customer would like to purchase a product/service, he/she can make the purchase within the chatbot, including the payment process. Bots, like 1-800flowers.com, eBay, and Fynd have already proved that.

—Julien Blancher, Co-Founder @ Recast.AI

In an article by Stefan Kojouharov, founder of ChatbotsLife, he mentions how different companies are making more money than they would have without chatbots. He says, The e-commerce space has begun using chatbots in a number of ways that are quickly adding dollars to their bottom line. Let's look at the early success stories:

- **1-800-Flowers:** reported that more than 70 percent of its Messenger orders derived from new customers!
- **Sephora:** increased their makeover appointments by 11 percent via their Facebook Messenger chatbot.
- **Nitro Café:** increased sales by 20 percent with their Messenger chatbot, which was designed for easy ordering, direct payments, and instant two-way communication.
- **Sun's Soccer:** chatbots drove nearly 50 percent of its users back to their site throughout specific soccer coverage; 43 percent of chatbot subscribers clicked through during their best period.
- **Asos:** increased orders by 300 percent using Messenger chatbots and got a 250 percent return on Spend while reaching 3.5 times more people.

Figure 1-2 tries to give you an idea of why there is a direct correlation between chatbots and revenue. Lets have a look at Figure 1-2 to get some idea about that.



Figure 1-2. Chatbot brings revenue

A Glimpse of Chatbot Usage

We will try to look at how useful chatbot has been for consumers due to its usability and the efficiency it provides. Everybody in this burning IT age wants to be fast in everything, and using chatbots makes your jobs easier and faster every day. It is personalized in a way as to not repeat obvious things; this makes us re-think about traditional usage of software. Figure 1-3 provides an illustration that should give you a fair idea about chatbot usage.



Figure 1-3. A glimpse of Chatbot usage by consumers

Customers Prefer Chatbots

Chatbots are not just software in the modern era. Chatbots are like our personal assistants who understand us and can be microconfigured. They remember our likes and dislikes and never tend to disappoint us by forgetting what we taught them already, and this is the reason why everyone loves chatbot. Next time you meet a person or meet your customer, don't forget to ask if they prefer conventional software or the new cutting-edge chatbots. Lets have a look at Figure 1-4 to understand the reasons why customers prefer chatbots compared to other software systems for human computer interactions.

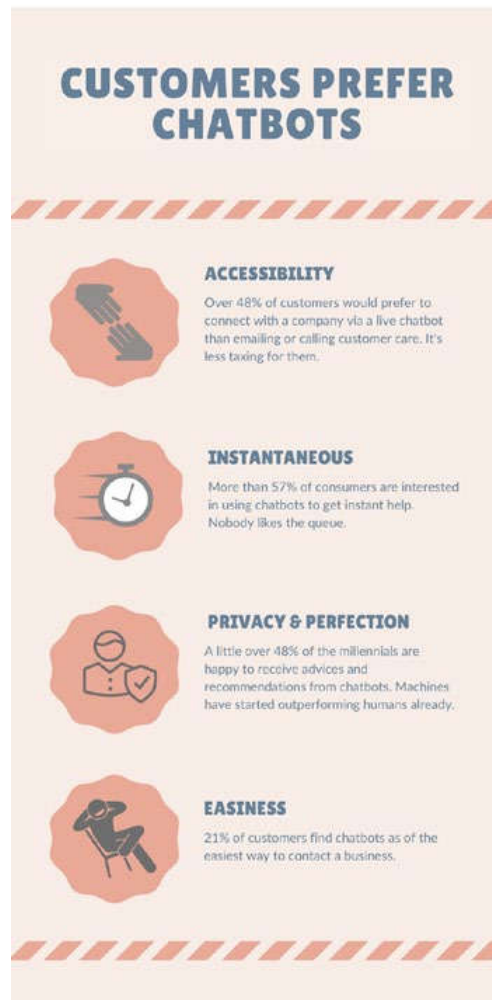


Figure 1-4. Customers prefer chatbots

In the next section of this chapter we are going to discuss why chatbots are the next big thing for budding developers. Whether you are a newer or a mid-level developer or an experienced SME you must understand what is available to developers when building chatbots.

The Developer's Perspective

Have you ever felt the pain when you have to update the OS of your computer or phone or any other app that you might be using in order to use new features? What if there is not much need to update the app every time to use new features? Or say, instead of having multiple apps, one could have one single app that did most of the things currently done by multiple apps?

Bots for developers are fun to build. It's like teaching your kid to walk, talk, behave, and do things. You love making it more intelligent and self-sufficient. From a developer's perspective, chatbots are a very important subject to know about.

Feature Releases and Bug Fixes

Lots of features can be added to the chatbot painlessly without having your users update your chatbot app. It might be a pain in the neck if you released a version of the app with some bug, and you have to fix it and release again in the AppStore for approval, and, most importantly, the users will have to update the app after all. If they don't update, then the customer will keep complaining about the issue, which results in productivity loss for everyone. In chatbots, everything is API-based, so you just fix the issue in the backend, deploy the changes in PRODUCTION, and woaah—issue fixed for your users without any worry. You save lots of time from user-reported bugs as well.

Imagine you built a bot to find restaurants and later you wanted to add the capability of searching for hotels, flights, etc. Users can easily just request such information, and your backend chatbot system will take care of everything.

Suppose you are building a Facebook Messenger chatbot; you can control almost everything, including what interface the user sees in his app, directly from your backend. In Facebook Messenger bots, you can choose whether the user gets to click on a button to say Yes/No or just enters simple text.

Market Demand

Fifty-four percent of the developers worldwide worked on chatbots for the first time in 2016. There is a huge demand for building a simple chatbot that works for companies, and they are looking for developers who can build it for them. Once you have completed Chapter 3 of this book I bet you can start selling your services to companies easily. You can also do your own startup in an area of your expertise by introducing a chatbot for that domain. Being able to build a chatbot end-to-end is a new skill to have, and that's the reason average market pay is also very good for chatbot developers.

The growing demand for chatbots can be seen in the number of chatbots being developed on developer platforms like Facebook. Facebook has 100,000 monthly active bots on the Messenger platform, and counting. You will be amazed to know that Messenger had 600 million users in April 2015, growing to 900 million in June 2016, 1 billion in July 2016, and 1.2 billion in April 2017.

Learning Curve

Whether you are from a frontend/backend background or know very little programming, there is immense possibility to learn new things when you are building or learning to build a chatbot. In this process you will learn about many things. For example, you get to learn more about Human Computer Interaction (HCI), which talks about the design and use of computer technology, focused on the interfaces between people and computers. You will be learning how to build or use APIs or web services, using third-party APIs like Google APIs, Twitter APIs, Uber APIs, etc. You will have immense opportunity to learn about Natural Language Processing, machine learning, consumer behavior, and many other technical and non-technical things.

Industries Impacted by Chatbots

Let's have a quick look at the industries that will benefit most from chatbots. A research study by Mindbrowser in association with *Chatbots Journal* collected data from 300+ individuals who participated from a wide array of industries including online retail, aviation, logistics, supply chain, e-commerce, hospitality, education, technology, manufacturing, and marketing & advertising. If we look at the chart in Figure 1-5, it is pretty much evident that e-commerce, insurance, healthcare, and retail are the industries benefiting most from chatbots. These industries rely heavily upon the responsiveness of the customer care team in an efficient manner that saves time. Given the fact that chatbot is good at that, it is evident that it will hail in these industries pretty quickly.

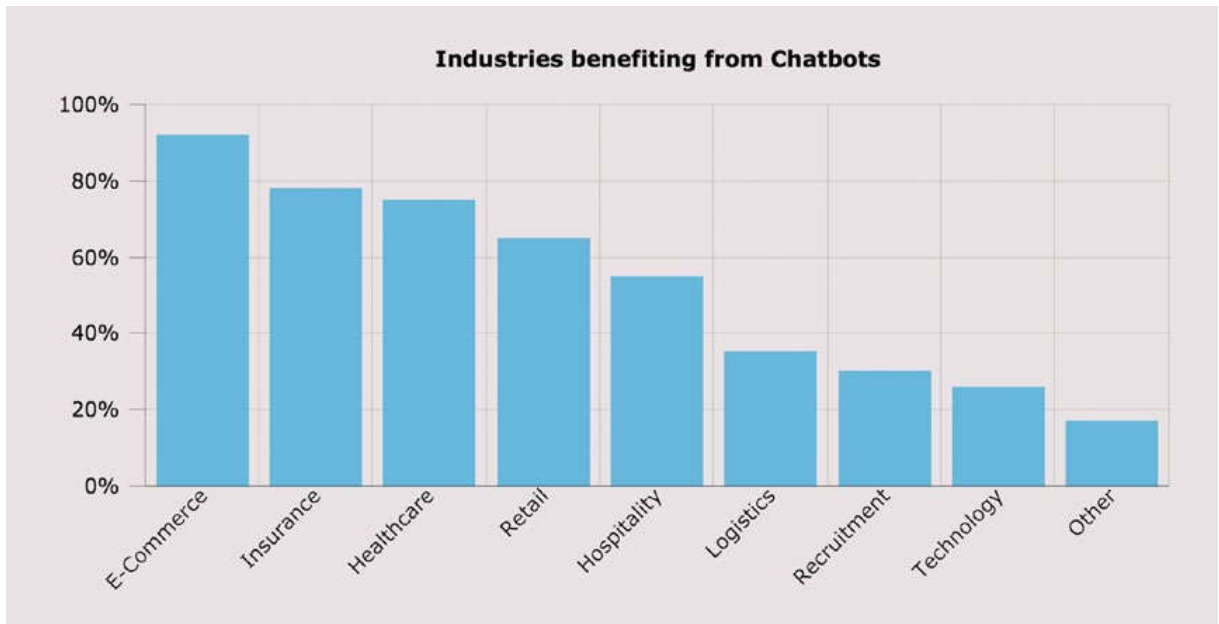


Figure 1-5. *Top industries that will benefit most from chatbots*

At this point of time, the chatbots are still getting traction in newer sectors in different forms. The next 5 to 10 years will be very much crucial for chatbots to spread the word in different industries that have no experience working with chatbots.

Brief Timeline of Chatbots

Let's look at the brief history of the timeline of how chatbots were formulated. It's very important to know where chatbot technology came from and how it was shaped. Chatbots have certainly gained popularity recently but the efforts are being made using decades of work with this technology. The history of chatbots will certainly amaze you regarding how far we have come since we started.

1950

The Turing test was developed by Alan Turing. It was a test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human.

1966

Eliza, the first chatbot, was created by Joseph Weizenbaum, designed to be a therapist. It used to simulate a conversation by using a “pattern matching” and substitution methodology that gave users an illusion of understanding on the part of the bot.

1972

Parry, a computer program by psychiatrist and Stanford scientist Kenneth Colby, modeled the behavior of a paranoid schizophrenic.

1981

The Jabberwocky chatbot was created by British programmer Rollo Carpenter. It started in 1981 and launched on internet in 1997.

The aim of this chatbot was to “simulate natural human chat in an interesting, entertaining and humorous manner.”

1985

The wireless robot toy, Tomy Chatbot, repeats any message recorded on its tape.

1992

Dr. Sbaitso, a chatbot created by Creative Labs for MS-DOS, “conversed” with the user as if it were a psychologist in a digitized voice. Repeated swearing and malformed input from the users caused Dr. Sbaitso to “break down” in a “PARITY ERROR” before it could reset itself.

1995

A.L.I.C.E (Artificial Linguistic Internet Computer Entity) was developed by Nobel Prize winner Richard Wallace.

1996

Hex, developed by Jason Hutchens, was based on Eliza and won the Loebner Prize in 1996.

2001

Smarterchild, an intelligent bot developed by ActiveBuddy, was widely distributed across global instance messaging and SMS networks. The original implementation quickly grew to provide instant access to news, weather, stock information, movie times, yellow pages listings, and detailed sports data, as well as a variety of tools (personal assistant, calculators, translator, etc.).

2006

The idea of Watson was coined from a dinner table; it was being designed to compete on the TV show “Jeopardy.” In its first pass it could only get about 15 percent of answers correct, but later Watson was able to beat human contestants on a regular basis.

2010

Siri, an intelligent personal assistant, was launched as an iPhone app and then integrated as a part of the iOS. Siri is a spin-out from the SRI International Artificial Intelligence Center. Its speech recognition engine was provided by Nuance Communications, and Siri uses advanced machine learning technologies to function.

2012

Google launched the Google Now chatbot. It was originally codenamed “Majel” after Majel Barrett, the wife of Gene Roddenberry and the voice of computer systems in the Star Trek franchise; it was also codenamed as “assistant.”

2014

Amazon released Alexa. The word “Alexa” has a hard consonant with the X, and therefore it can be recognized with higher precision. This was the primary reason Amazon chose this name.

2015

Cortana, a virtual assistant created by Microsoft. Cortana can set reminders, recognize natural voice, and answer questions using information from the Bing search engine. It was named after a fictional artificial intelligence character in the Halo video game series.

2016

In April 2016, Facebook announced a bot platform for Messenger, including APIs to build chatbots to interact with users. Later enhancements done included bots being able to participate in groups, preview screens, and QR scan capability through Messenger's camera functionality to take users directly to the bot.

In May 2016, Google unveiled its Amazon Echo competitor voice-enabled bot called Google Home at the company's developer conference. It enables users to speak voice commands to interact with various services.

2017

Woebot is an automated conversational agent that helps you monitor mood, learn about yourself, and makes you feel better. Woebot uses a combination of NLP techniques, psychological expertise (**Cognitive-behavioral therapy** [CBT]), excellent writing, and a sense of humor to treat depression.

What Kind of Problems Can I Solve Using Chatbots?

This question becomes challenging when you don't know the scope of your bot or don't want to limit it to answer queries.

It's very important to remember that there is a limit to what chatbots can do. It always feels that we are talking to a human-like thing that is very intelligent, but the specific bot is designed and trained to behave in a certain way and solve a specific problem only. It cannot do everything, at least as of now. The future is definitely bright.

So, we come to the question of finding out if your problem statement is really good to go and you can build a bot around it.

If the answer to all of these three questions is yes, then you are good to go.

Can the Problem be Solved by Simple Question and Answer or Back-and-Forth Communication?

It's really important to not try to be a hero when solving any problem that is very new to you. You should always aim to keep the problem scope limited. Build the basic functionality and then add on top of it. Don't try to make it complex in the first cut itself. It doesn't work in software.

Imagine Mark Zuckerberg thinking out loud and spending time building all the features of Facebook at the start. Tagging a friend, having a like button, liking a user comment, better messaging, live video, reactions on comments, etc.—these features didn't exist even when Facebook was funded with over 1 million registered users on the platform. Would he have really succeeded if he would have gone on to first build these features and then launch the platform?

So, we should always try to create features only needed at the moment without having to over-engineer things.

Now, coming back to the first question, “Can the problem be solved by simple question and answer or back-and-forth communication?”

You just have to keep your scope limited and your answer will be yes. We are not at all limiting ourselves to solving complex problems but definitely limiting ourselves to solving a complex problem all in one go.

“You have to make every single detail perfect. And you have to limit the number of details.”

—Jack Dorsey

Does It Have Highly Repetitive Issues That Require Either Analyzing or Fetching of Data?

This question is important because either from a business perspective or a developer's perspective, what chatbot does and is made to do is to make people using it efficient and productive. And how do you do that? By removing the need of a user to do repetitive things themselves.

Chatbots are definitely more capable of just automating some highly repetitive stuff, but you will always find that most of the chatbots primarily try to solve the same issue—be it by learning under supervision (read: “By Supervised Learning”) or self-teaching (read: “By Un-supervised Learning”).

Can Your Bot's Task be Automated and Fixed?

Unless you are thinking of building a chatbot just for your learning purpose, you should make sure the problem you are trying to solve can be automated. Machines have started to learn and do things themselves, but still it's a very nascent stage. What you think can't be automated now may be automated in a few years.

A QnA Bot

One of the good examples of a problem statement for building a chatbot could be a QnA Bot. Imagine a bot that is trained to understand various user questions whose answers are already available on an FAQ page of a website.

If you go back and try to find the answer of the aforementioned three questions, the answer will be yes.

See Figure 1-6 and you will find what an FAQ bot is doing.

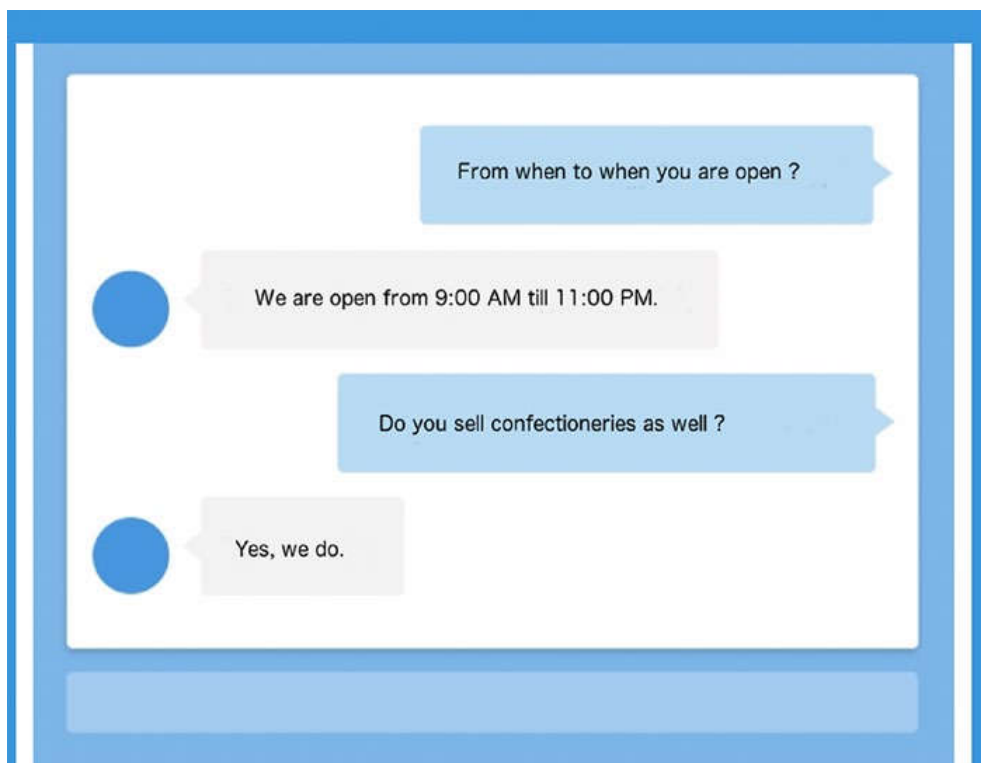


Figure 1-6. FAQ chatbot example

These are nothing but very repetitive questions that customers of a particular store may call and ask or try to find answers to by going to a website and navigating through the pages.

Think when you have a chatbot like this and it answers your question like a human in seconds and even does more than you could imagine. This is just a little of what chatbots are capable of.

Now, let's try to analyze the aforementioned three questions and their answers in case of QnA Bot.

- Can the problem be solved by simple question and answer or back-and-forth communication?

Yes, FAQs are nothing but simple frequently asked questions and their relative answers. There may be a context-based FAQ, but unless you are solving a multidomain problem using chatbots, you won't be having this problem. There could be a situation where two or more questions may seem similar, but you can always design the bot to ask a question back to the user when it's doubtful.

- Does it have highly repetitive issues that require either analyzing or fetching of data?

Yes, FAQs require us to fetch the data from the database and show it all at once in the website or possibly dynamically. But the user has to go through all questions one by one to find the question he/she is looking for and then see its answer. Lots of combing through the UI before the consumer actually gets his/her answer... or maybe not. Why not let our bot do that for us?

- Can your bot's task be automated and fixed?

Yes, an FAQ bot would need to get the question, analyze the question, fetch information from the database, and give it back to the user. There is nothing here that can't be done using coding. And also, the process is pretty much fixed won't change in real-time.

Starting With Chatbots

There are three steps one should follow before building chatbots. We'll discuss each one of them briefly here.

1. Think about all the scenarios or tasks you want your chatbot to be able to do, and gather all related questions in different forms that can be asked to do those tasks. Every task that you want your chatbot to do will define an **intent**.
2. Each question that you list or intent can be represented in multiple ways. It depends on how the user expresses it.

For example: Alexa, Switch off the light. Alexa, Would you please switch off the light? Can you please switch off the light? A user may use any of these sentences to instruct the bot to switch off the light. All of these have the same intent/task to switch off the light, but they are being asked in different **utterances/variances**.

3. Write all your logic to keep the user tied to the flow that you have chosen after you recognize the user's intent.

For example, suppose you are building a bot to book a doctor's appointment. Then you ask your user to give a phone number, name, and specialist, and then you show the slots and then book it.

In this case you can expect the user to know these details and not try to accommodate all the things in the bot itself, like a specialist for an ear problem is called an ENT. However, doing this is not a big deal. So, again it comes back to deciding the scope of your bot, depending on the time and resource you have to build the application.

Decision Trees in Chatbots

If you know about [decision trees](#), then that's very good because you will be needing that knowledge frequently when designing the flow of your chatbots. But if you don't know about the decision trees, then just Googling would help you learn this simple concept widely used in Computer Science.

Using Decision Trees in Chatbots

In the context of chatbots, a decision tree simply assists us in finding the exact answer to a user's question.

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

—Wikipedia

The most difficult part when building a chatbot is to keep track of if...else code blocks. The greater the number of decisions to make, the more frequently if...else comes up in the code. But at the same time these blocks are required to encode the complex conversational flows. If the problem is complex and requires a lot of if...else in real-life, then that will require code to adjust in the same way.

How Does a Decision Tree Help?

Decision trees are simple to write and understand, but they are a powerful representation of the solution made for the problem in question. They inherit a unique capability to help us understand a lot of things.

- Help in creating a full picture of the problem at hand. Looking at the decision tree, we can easily understand what's missing or what needs to be modified.
- Helps debug faster. Decision trees are like a short bible or, say, a visual representation of a software requirement specification document, which can be referred by developers, product managers, or leadership to explain the expected behavior or make any changes if needed.
- AI is still not at that stage that it can be trained with lots of data and perform with 100 percent accuracy. It still requires a lot of hand-holding by writing business logic and rules. Decision trees help wherever it becomes a little tough to ask a machine to learn and do it.

Let's take a simple example and try to understand how it helps in building chatbots. Look at the example diagram for a chatbot that starts with a question of whether the user is looking for a t-shirt or jeans, and based on the input the diagram flow goes further to give options related to the product by asking more questions. You don't need to create a full-fledged decision tree, but you should definitely have a flow of questions defined at every step before starting to build chatbots.

Suppose you were building a similar chatbot that helps people buy apparel online. The first thing you would do is to make a similar decision tree or a flowchart to help your chatbot ask appropriate questions at the right time. This is really needed to set the scope of each step and what needs to be done at that stage. You will need the state diagrams or a simple flowchart later when you actually code your first chatbot. Remember to not be too stringent while creating a diagram like Figure 1-7; keep it as simple as possible and then add the extended functionalities later. The benefit of such a process is the development time will be cut down, and later on the functionality will be loosely coupled and would start making sense as components. Like in the example, after creating the basic functionality, you can add color choices, price range, ratings, and discount options as well.

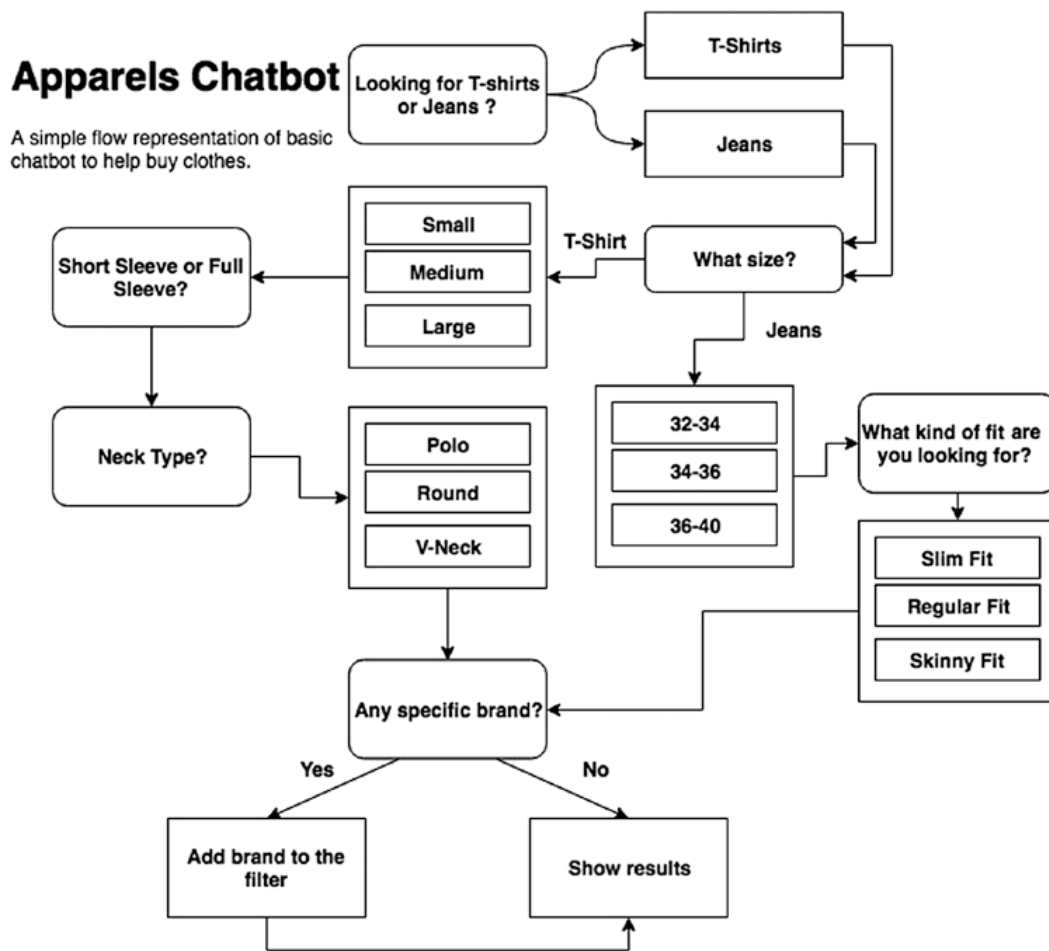


Figure 1-7. A simple representation of an apparel chatbot for buying clothes online

There are definitely more things you can add to the earlier use-case depending upon your requirements. But you have to make sure that you don't make it too complex for yourself as well as for the user.

A decision tree not only helps you to keep the user tied to the flow but also is a very effective way to identify the next intent that might be coming in the form of a question from the customer.

So, your bot will ask a series of questions following the decision tree that you have built. Each node narrows down on the customer's goal through chatbot intents.

Suppose you were creating a chatbot for a financial institution—say, a bank—that can do a money transfer based on your request after authentication. In this case, your bot may first want to verify the account details and ask the user to confirm the amount, and then the bot may ask to validate target account name, account number, account type, etc.

You cannot or would not want to invoke an OTP (one-time password) API unless you have validated if the user’s account balance is more than the requested amount.

It happens with all of us and with customers as well. They get frustrated when their questions are not answered correctly. Using decision trees for your chatbot will definitely make the experience better for your users than it would be if your didn’t use them.

Lots of times you will find issues solving some intents programmatically. So, the bottom line is, *“If you can’t solve something programmatically then solve it by design.”*

Look at Figure 1-8 where the bot is trying to take a health quiz and wants to know if antibiotics can work for everything.

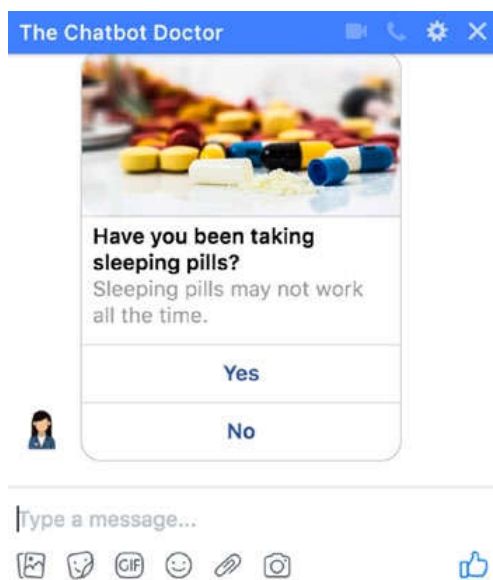


Figure 1-8. Example of solving a use-case by design

Since the answer is expected to be a Boolean (True/False), you give just two buttons for the user to click instead of letting them type and wait to fix their mistake.

This is solving by design rather than writing lots of code that will be handling unexpected user inputs. You will have so many scenarios while building the chatbots where by just giving buttons, you will be able to quickly know the intent of the user. It’s important to understand such scenarios and provide buttons both for your own convenience as well as for users who don’t need to type in obvious cases of optional answers.

The Best Chatbots/Bot Frameworks

- <https://woebot.io/>
 - Can track your mood
 - Helps you feel better
 - Gives you insights by seeing your mood pattern
 - Teaches you how to be positive and high-energy
- <https://qnamaker.ai/>
 - Build, train, and publish a simple question-and-answer bot based on FAQ, URLs, and structured documents in minutes.
 - Test and refine responses using a familiar chat interface.
- <https://dialogflow.com/>
 - Formerly known as api.ai and widely popular among chatbot enthusiasts.
 - Give users new ways to interact with your product by building engaging voice-and text-based conversational interfaces powered by AI.
 - Connect with users on the Google Assistant, Amazon Alexa, Facebook Messenger, and other popular platforms and devices.
 - Analyzes and understands the user's intent to help you respond in the most useful way.
- <https://core.rasa.ai>
 - A framework for building conversational software
 - You can implement the actions your bot can take in Python code.
 - Rather than a bunch of if...else statements, the logic of your bot is based on a probabilistic model trained on example conversations.

- <https://wit.ai>
 - Wit.ai makes it easy for developers to build applications and devices that you can talk or text to.
 - Acquired by Facebook within 21 months of its launch, wit.ai team contributes toward Facebook's own NLP engine inside Facebook.
 - You can use wit.ai for building chatbots, home automation, etc.
 - Wit.ai is similar to the way Dialogflow works but is not as feature-rich as Dialogflow. People initially used wit.ai, as it was free and Dialogflow was not, but later on Dialogflow became free as well.
- <https://www.luis.ai/>
 - A machine learning-based service to build natural language into apps, bots, and IoT devices.
 - Quickly create enterprise-ready, custom models that continuously improve.
- <http://botkit.ai>
 - Visual conversation builder
 - Built-in stats and metrics
 - Can be easily integrated with Facebook, Microsoft, IBM Watson, Slack, Telegram, etc.

Components of a Chatbot and Terminologies Used

Components of a chatbot system are very few. In this section we'll be briefly discussing the components of a chatbot that you will come across in the later chapters.

Having a basic theoretical understanding of any system before diving deep is always helpful. You should have a fair idea after this section about technical terminologies used while building chatbots using Python. These terminologies will be used frequently in coming chapters when we actually start building our chatbots.

Intent

When a user interacts with a chatbot, what is his intention to use the chatbot/what is he asking for?

For example, when a user says, “Book a movie ticket,” to a chatbot, we as humans can understand that the user wants to book a movie ticket. This is intent for a bot. It could be named “*book_movie*” intent.

Another example could be when a user says, “I want to order food,” or “Can you help me order food?” These could be named “*order_food*” intent. Likewise, you can define as many intents as you want.

Entities

Intents have metadata about the intent called “**Entities.**” In the example, “Book a movie ticket,” booking a ticket could be an intent and the entity is “**movie,**” which could have been something else as well, like flight, concert, etc.

You can have general entities labeled for use throughout the intents. Entities could represent as a quantity, count, or volume. Intents can have multiple entities as well.

For example: Order me a shoe of size 8.

There could be two entities here:

Category: Shoe

Size: 8

Utterances

Utterances are nothing but different forms of the same question/intent your user may show.

- Remember we discussed the switching off the light intent? That was an example of how a user can use different utterances for the same intent.
- It is suggested to have an optimum 10 utterances per intent and a minimum of 5, but this is not restricted.

Training the Bot

Training essentially means to build a model that will learn from the existing set of defined intents/entities and utterances on how to categorize the new utterances and provide a confidence score along with it.

When we train the system using utterances, this is called supervised learning. We will soon be learning more about doing this practically.

Confidence Score

Every time you try to find what intent an utterance may belong to, your model will come up with a confidence score. This score tells you how confident your machine learning model is about recognizing the intent of the user.

That's all we wanted to cover in the first chapter of "Introduction to Chatbots." You must have a fair idea about chatbots from a business perspective and from a technical perspective. We walked through the lane of history belonging to chatbots. It's quite fascinating how far chatbots have evolved.

We learned about how chatbots have evolved over a period of time and why chatbots are a must for a business to grow in this cutthroat competition. We learned about the different chatbot frameworks and also got to know about the terminology used for chatbots by example. We'll be using them in the coming chapters. You should be at a stage now where you know what kind of chatbot you want to build and how it would behave when built.

Do all your write-ups and decision trees, if needed, and after we have learned the basics of Natural Language Understanding in the next chapter, we can quickly start building our chatbot.

Don't worry even if you don't have anything in mind. We'll try to build a cool chatbot step by step with all the concepts learned in the upcoming chapters.

See you in the next chapter.