

10 Steps to Chatbot Creation

So, you want to build a chatbot, great. You're not alone. There are more than 34,000 chatbots on Facebook Messenger alone, and many of these are already built by and for brands. Increasingly, they are helping customers answer questions and solve issues on their own as well as taking work off the shoulders of contact center agents. But even though you're not too late to the game, you might be asking yourself: Where to begin?

The following instructions are designed to guide you through the vital steps necessary to build a customer service chatbot from scratch. It will show you how to find the right use case, design both the message understanding and the message (response) generation side of the bot, pick the right tools and approaches for testing, and deploying your bot. This is not meant as an exhaustive guide, but should help you get started with the planning and thought process around your own chatbot.

1. Determine the Role of the Bot and Set Goals

Before you start anything, identify what you want your bot to do and be specific. Identify all required features and address the implementation tasks. Doing this upfront will lead to a robust chatbot and ideally one that can deal with the majority of topics your contact center handles today, including strategies on how to handle unknown answers to questions.

To start, consider a modest goal that matches your customer service philosophy. Is the initial plan to provide great responses to a limited set of questions with high confidence or to handle a broader base of topics to satisfy customer needs? Eventually, both goals can be met but there will be learning and refining which will happen once a chatbot is available for your customers to use.

Let's begin with a simple (and friendly) bot whose purpose is solely to route questions to proper live staff. This is the equivalent of a contact center agent's first day in your contact center – both will be able to handle more inquiries over time. Start collecting customer data without answering any questions.

* Identify the top 5 to 10 items that customers want to know about. Limiting the topics allows for finer tuning of the bot to understand the subject better which in turn allows answers with higher confidence.



* Construct Frequently Asked Questions responses (FAQs). These are normally 10 to 25 topics and answers with short text responses along with follow up links as needed which contains additional details and information. However, note that the chat medium demands a more conversational approach, so make sure responses are less regurgitation of long explanations, and more chat.

* Match the chatbot to your existing voice self-service platform capabilities. Recreating well-defined and understood topics creates consistency across all modalities for your customers.

* Identify the transactional use cases beyond merely informational ones. This would be for creating a more fully functional bot that manages all types of the business tasks your staff handles today, such as providing detailed order status or handling appointment reminders.

* No matter which direction is decided upon to get started, it is important that the bot-customer transcript be provided to the agent so they have context to help the customer and provide a higher touch experience.

~ Bill Gay, Director Self Service

2. Evaluate and Pick a Channel

Text-based chatbots can live on any communication channel that can carry a dialog, whether that's a traditional mobile carrier channel (SMS, USSD), a messaging app (Facebook Messenger, WeChat, Kik, Line, Viber), certain social networks like Twitter, or chat embedded on a website. If you are considering adding a conversational experience to your existing mobile app, ask this question: does this experience add enough new value that it justifies the investment? App fatigue is starting to reduce app download and engagement rates and can have a like-effect on the use of in-app chat.



Whatever channel you prefer, make sure it offers an open API so that bots can be programmatically embedded. That is not yet the case for some of the messaging apps such as WhatsApp (as of February, 2017).

The channel you choose can be one you already engage customers on, or it could be completely new to you. The latter has the advantage that it wouldn't conflict or compete with any existing programs in your organization, and you can start fresh. Oftentimes, launching pilot implementations is easier on a new channel vs. one you already use. ~ *Tobias Goebel, Director of Emerging Technologies*

3. Create the Conversational Architecture



4. Design Dialog Flows and Storyboards.

While creating the Conversational Architecture is an exercise to help you organize your content and start thinking about the best ways to word the bot's answers, the dialog flow goes into all the detail needed for your developer to implement the bot.

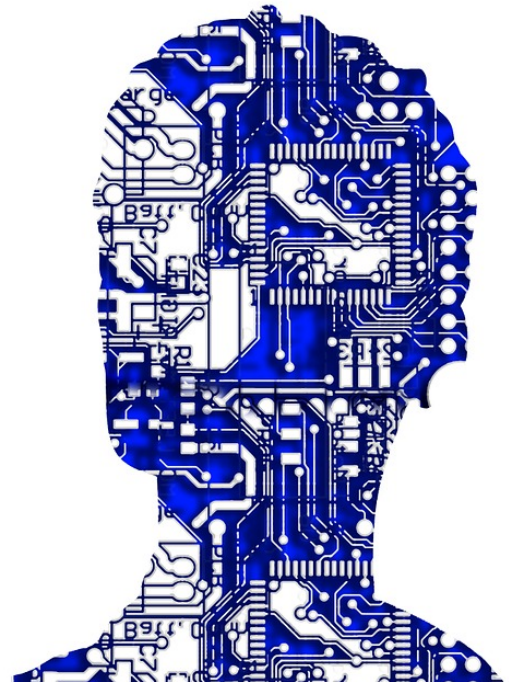
In the dialog flow design, you will want to show representations of what the bot will say at each step. However, the detailed message design should happen outside of the flow diagrams, as you will want to design variations of the same message for frequently occurring dialog steps. This is a technique called **random prompting**, where you make the bot use wording variations to essentially say the same thing. This will make the whole experience less robotic and more human-like, something we should strive for in bot design. As an example, consider designing multiple ways of saying "Not sure I got that, can you rephrase please?", since so-called "NoMatches" (i.e. the bot cannot find a match of the customer message against its rules) can occur frequently, especially in the early stages of your bot project. ~ Tobias Goebel, Director of Emerging Technologies

5. Design the Integrations

Integrations can be as varied as the applications you want to automate. Are there already current integrations and/or data that can be used to support the customer experience with your bot? If you already have an integration with your existing self-service platforms (web or voice) to some of your systems, the same integrations may be re-used for chatbot purposes. For example, if your IVR application already verifies your customer and can provide order status information, then most likely the integration can be re-used for your chatbot.

New bot self-service concepts however may require new integrations. Current software packages, technologies, and databases generally have robust connections that can provide a wealth of detail required for the bot to function.

Accessing customer data to answer customer questions is important, but not all chatbot functions require integration. For example, when FAQ answers are well-defined and documented, they can be re-used to build bot responses in order to provide message consistency. This example does not require any back-end integration to other systems to get started. Down the road, you might want to better manage answers to FAQs centrally, which will require access to a Knowledge Base or similar systems. ~ Bill Gay, Director Self Service



6. Collect Chat Data

No matter what platform you use to create your bot, one of the most important resources you will need is a collection of sentences reflecting different ways to express each one of the intents and slot values your bot will need to recognize. This resource can be difficult to create because of the amazing diversity of human language – and the limitations of the developer's imagination. The ideal resource is actual text from conversations with customers. If you're lucky enough to have had agents communicating with customers in chat channels, you can mine those conversations for real examples (see also the next step). But for many companies, this may be the first time they are engaging in customer service through a chat-based interface.

If you need to create your initial data by hand, make certain that your collection includes not just one developer's intuitions but also input from as large a variety of people. Remember that diversity is the key. One useful approach is therefore to work with QA and crowdsourcing companies who have access to large number of people from around the world. Finding people with the same linguistic background as your target users is helpful. e.g. if the target language for your bot is English, but you know that your users will come from different countries, finding people that have the same linguistic background helps create more realistic data; data that might even include mistakes that are typical for some non-native speakers.

For FAQ-type bots, an approach that can be used for this exercise is a “Jeopardy!”-like approach: provide your designers with just the various answers your bot will give, and ask them to brainstorm questions that would result in these answers. ~ *Lisa Michaud, NLP Architect*

7. Pick a Platform and a Development Approach

In essence, most chatbots consider the key tasks to be performed on natural language sentences to be: (1) to determine the intent of the sentence (what is the customer asking, or requesting? What use case does she wish to initiate?) and (2) to extract data from the sentence (what options has the customer requested? What data is he providing to you?). There are essentially two different approaches to these tasks: one based on explicitly creating rules from the top down, and one using machine learning algorithms to learn the task from a large corpus (a collection of written texts) of transcribed interactions.

If you have been offering web chat for quite some time, you might already have collected tens of thousands of customer inquiries with corresponding answers. While a true dialog consists of more than one turn, and many customer service dialogs are not simple pairs of question & answer, you should be able to apply machine learning algorithms to this data set to learn the most common answers to the most common questions. Note that with this approach, you will have to start from scratch for every new language you want the bot to speak. Also, it is still a tremendous and largely manual effort to tag the data and analyze the outcome to ensure quality.

If you don't have such a corpus already available to you, or what you have is not suitable to train an algorithm, you will have better luck with an approach based on writing distinct rules to extract meaning from messages. The easiest form of doing this is to find certain keywords in the customer's message and act upon that. However, that bears a risk in that many messages could be misclassified if keywords don't necessarily appear in the form expected. Consider the difference between "Can I book a flight for tomorrow" and "Can I read an electronic book on my flight tomorrow?" for an airline chatbot, which are asking for two different things yet both contain the keywords "book", "flight", and even "tomorrow." This might make a simplistic rule believe that both messages are about booking a flight for tomorrow. The better platforms out there extend the reach of a rules-based approach with built-in linguistic tools that can leverage the relationships between words (synonyms, hypernyms/hyponyms, domains) or common syntactic patterns, in many different languages. This makes it easier both to capture broad linguistic variation in concise rules and to distinguish between senses of a word like “book.”

An advantage of the linguistic top-down approach is that you have full control over how a message is understood. A neural network created by a machine learning algorithm is often a black box that doesn't let you go in and surgically change how one particular message is understood. Nuances in natural language – such as the fact that “I want to transfer my data” and “How do I move my files?” are about the same intent, but “How do I move this to my file?” is from a different one – are hard to learn with machine learning, but easier to distinguish with a linguistics-based platform. ~ *Tobias Goebel, Director of Emerging Technologies and Lisa Michaud, NLP Architect*

8. Implement the Dialogue Flow and Engineer the NLU

It all comes together in this step: the conversational architecture, the dialogue flow and storyboard, the platform you have selected, and the data you have collected. Your essential task is to use these to create a classifier that will map an incoming text to the system's response.

If you selected a platform based on machine learning, you will provide this platform with your example sentences for each possible intent. The more examples you provide, the better the algorithm will learn the variations of linguistic expressions that can be used for each intent, and the better it will learn how to distinguish between intents. Note that you will want to reserve some of your example sentences for the next step (testing).

If you are working with a linguistic rules-based platform, you will use the sentences in a different way. The rules you craft will explicitly represent the characteristics that determine that a given sentence belongs to intent A or intent B, leveraging the tools and abstractions mentioned earlier.

In either situation, this is when it is centrally important to have a diverse set of examples as close to real user utterances as possible. A single source or a small set of sources will not begin to capture the universe of

differences your real users will display when they express themselves to your system. ~ *Lisa Michaud, NLP Architect*

9. Internal Testing and Revision of Your Use Case Detection

Now you're ready for the second use of your corpus of example sentences: automated testing. You also want as many diverse human testers as possible for "real user" testing. Test and revise your NLU component as well as the bot flow until you reach an acceptable level of accuracy. Note that this step and the step previous to it are **iterative** and **approximative**; because of the nature of human language and the infinite possible expressions of every intent, the goal of 100% accuracy is an unattainable one. Each time you iterate through these steps, however, you get closer.

Furthermore, as mentioned before, make sure you get as close as possible to your real end-users. Any tester that isn't a real user of your system with an honest need to chat with your bot will produce results that are slightly artificial in nature. You might want to consider a smaller rollout with a fraction of your target customer base to vet some of the design decisions you've made along the way. ~ *Lisa Michaud, NLP Architect*

10. Early Deployment and Revisions

Even though you're ready to go live, the work is not done when the bot gets deployed. Even if your bot employs some kind of unsupervised or semi-supervised learning to adjust its own behavior over time, monitoring the first interactions with real users will yield very useful information and may signal that explicit adjustments should be made. Typical adjustments are in the wording of your bot's responses, as they might yield follow-up clarification questions by your customers that wouldn't be necessary if the bot's answer were clearer. You may need to adjust the logic of your intent classification, either through explicit manipulation of the rules or through providing more example sentences. Finally, you may need to add new use cases if the designed use cases do not cover the majority of user requests. If you truly started small as recommended, then this is the time when you are collecting the vital information about which use cases are the key ones to cover. ~ *Lisa Michaud, NLP Architect*

To ensure a successful outcome of your chatbot deployment, view the creation as an iterative process: gather the data, review it, and apply it to your bot's design. Repeat. Above all, log **everything** for the future. The success of other projects could be driven by the lessons you learn from this one.