

# A developer's guide to chatbots

Michael Yuan

August 10, 2016

One of the current trends in technology is the chatbot. This tutorial summarizes the major messaging platforms, bot frameworks, and artificial intelligence (AI) services you use to develop your chatbot applications.

The very first chatbot, ELIZA, was developed 50 years ago at the Massachusetts Institute of Technology. It simulated a Rogerian psychotherapist, someone who just repeats the human user's words back to the human; it is not very good. In the following decades, chatbots were mostly of academic interest. But in recent years, smartphone-based chatbots have gained wide interest from the industry with high profile products such as the Apple Siri, Amazon Echo, and China's WeChat.

In 2016, chatbots are [one of the hottest trends in technology](#). The biggest messaging platforms, such as Facebook Messenger and Skype, announced developer programs to support chatbot applications. At the same time, smaller messaging platforms, such as Slack and Telegram, launched "bot stores" and investment funds to attract developers. Google is betting directly on a chatbot application (Allo) powered by its artificial intelligence (AI) and big data. As a developer, right now is the best time to jump in and surf the big wave of the "next big thing" in technology.

## The convergence of market forces

The rise of chatbots is the direct result of changing user behavior in the post-app mobile world and the maturing of key enabling technologies.

- After nearly a decade of explosive growth, mobile apps have largely [stopped growing](#). And as the *Wall Street Journal* says in "[What Comes After Apps](#)," smartphone users have reached their limits on how many apps they want to install and open on a daily basis.
- Users are spending an increasing amount of time in just a few apps. And social and [messaging apps emerge as big winners](#). Users do not like to exit their messaging app just to glance at small snippets of information (for example, to check the weather, stock prices, or look up a restaurant or map).
- Younger users grew up with computers and are much more computer savvy than older users. As a result, younger users are more comfortable with text messaging and communicating with computers by using commands, which have long been favored by power users over slower GUIs.

- Consumers, especially affluent consumers, are outsourcing their "chores," such as driving, shopping, cleaning, food delivery, and errands, giving rise to the "gig economy" and the need for more customer service.
- AI has gotten a lot better in the past couple of years in understanding what the user wants in natural language.
- The API economy has matured to the point where many useful services can be accessed from computers to complete real-world tasks.

In short, chatbots are used in human users' preferred environments (messaging apps), converse with users in natural language, understand what users want, and do the humans' bidding through a large network of connected services.

## Bot platforms

Now let's look at some messaging platforms where you can build bots.

Facebook Messenger is one of the most popular messaging platforms with more than 1 billion monthly active users. Within a few months of Facebook opening its bot platform to developers, over 23,000 developers were building bots. They have launched over 18,000 bots so far. Facebook Messenger bots are closely associated with Facebook pages, which is quite ubiquitous among businesses.

Kik messenger is a popular messaging application with over 200 million monthly active users. A key feature of Kik is its anonymity, which is also controversial given the large number of teenagers on the platform. Kik launched a bot store to promote bots on its platform in April 2016.

Telegram is a relatively new messaging application. It was launched in 2013 and now has 100 million monthly active users. A key differentiator of Telegram is its advanced security and encryption features. Telegram has supported bots since June 2015. Similar to Facebook Messenger, its bots display rich UI elements such as buttons and formatted cards.

Skype is a messaging platform with over 300 million monthly active users. Besides text messaging, Skype's traditional focus is on voice and video calls.

Twitter is a public messaging platform with approximately 300 million monthly active users. Bots have always been a part of Twitter as "non-human" users. By using the Twitter API, a bot can follow people and send out Tweets.

WeChat is the dominant mobile messaging platform in China and has over 700 million monthly active users. WeChat is a pioneer in chatbots, with a wildly successful bot platform. In China, businesses often have a WeChat bot long before they set up a website.

SMS and email are traditional messaging platforms with the greatest user reach. That said, the number of messages that are sent by using mobile apps such as Facebook Messenger has far exceeded SMS in recent years. You can use telecom APIs such as Twilio to build SMS bots, and email bots can be built on the open email protocols of the Internet or by using specialty API

providers such as lonelybots. The email platform already has useful chatbots, for example, the smart scheduling assistant x.ai bot.

Slack is a group messaging platform for work-related teams. As of February 2016, it had 2.3 million monthly active users, and it's growing fast. Slack is one the pioneers in the current wave of chatbot renaissance. Slack provides one of the first "bot stores" in the industry so that teams can discover and install bots easily. As expected, most Slack bots are work and productivity related.

Beyond mobile messaging applications, numerous messaging service providers also exist. Mobile apps integrate their services to provide in-app messaging capabilities that app users can use to message each other. Because those messaging services are API driven, it is possible to develop bots to interact directly with those APIs and chat with application users. Examples of such in-app messaging libraries and SDKs include EaseMob, Layer, Tiger Connect, and others.

## Bot frameworks

As a chatbot developer, it can feel overwhelming to develop and target so many messaging SDKs at the same time. Bot frameworks are software frameworks that abstract away much of the manual work that is involved in building chatbots. Many such offerings are on the market.

However, although many bot frameworks boast "write once deploy anywhere," you are more likely to create a separate chatbot for each of your target messaging platforms. This is because one-size-fits-all solutions must conform to the lowest common denominator of all the bot platforms they support. That often creates a less optimal user experience, especially in the early days of the ecosystem when the native bot platforms themselves are fast evolving and constantly adding new features.

Furthermore, framework solutions are not great for beginners to learn about chatbot development. They try to automate too much and obscure the underlying mechanism for starters. They constrain you to the UI features they support, but their innovation is at least one step behind the native messaging platforms. Finally, the existing frameworks are mostly based on node.js, while a vast majority of server-side developers are in more mature programming environments such as Java™ programming and PHP.

For effective learning, I propose an "open source lightweight framework" approach. It establishes a simple request- and response-based programming convention for all bots. The goal is to have "similar" code for each bot platform, not identical code to run everywhere. This way, you have the flexibility to add features specific to each bot platform, while the framework still abstracts away common boilerplate code such as security, logging, analytics, caching, and API integration. You can download and learn about my own lightweight framework in both the Java language and PHP from [GitHub](#).

## AI services

As discussed earlier in this tutorial, advancement of natural language AI services is a key enabling factor of chatbots. AI enables modern chatbots to escape the trap of a "telephone tree," and

become partners of human users. Four important types of AI services are related to natural language.

- Rule-based pattern recognition: Examples of this type include date, email, phone number, quantity, and trigger words. These rules are typically implemented as regular expressions in the application. The benefit of such rules is that they are precise, and you can add and subtract rules to handle new situations and address bugs with certainty.
- Natural language classifier: This AI service type is used to detect and classify intent of a user command. For example, when the user asks "Is it hot today?", the AI should know that the user is asking about today's temperature. This classifier is constructed by training an algorithm with many real-world speech utterances in an application area. An example of this is the Watson [Natural Language Classifier](#), a web service provided by IBM that you can use to train your model. After training, you can submit new sentences by using the web service, and the service will return the classifications with a confidence level for each.
- Rule-based conversation manager: Based on the user's intent and data that is associated with the intent (called entities), such as location and time, the service can apply rules and generate scripted responses. The idea behind this type of service is that most chatbot conversations are very similar. Therefore, they can be captured in scripts without the developer having to program a state machine with numerous IF-THEN statements.
- Speech recognition: Although many high-end smartphones today have speech recognition that is built right into their text input methods, third-party voice recognition is still important for many less expensive mobile or PC devices. In addition, it is important in some use cases when the user's voice might be stored for later analysis (such as in customer service when a supervisor might want to review the user's mood and tone). IBM Watson provides a [text-to-speech](#) service and a [speech-to-text](#) service in seven international languages.

**Note:** The [IBM Watson Developer Cloud](#) has a comprehensive collection of ready-to-use AI and NLP services for chatbot development and many other applications. For instance, the [Watson Conversation Service](#) combines NLP intent classification, entity detection, and scripted conversation management.

## Summary

This tutorial explained chatbot development platforms, developer frameworks, and AI services. To get started with chatbot development, I recommend starting with a popular messaging platform, and then download the lightweight framework source code from GitHub to play with it yourself.

© Copyright IBM Corporation 2016  
([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))

### Trademarks

([www.ibm.com/developerworks/ibm/trademarks/](http://www.ibm.com/developerworks/ibm/trademarks/))