CHATBOTS

The road to human intelligence

By Alberto García and Mikel San Sebastián

INDEX

1. Introduction

Before chatbots there were simply bots: The invention of a chatbot brought us to the new era of technology, the era of conversational services. A chatbot is a computer program designed to interact with users via textual or auditory methods using artificial intelligence. We can also call it a Personal digital assistant.

Chatbots are currently gaining a lot of popularity especially in the business sector as they have the potential to automate customer services and reduce human efforts. For a chatbot to perfectly emulate a human dialogue, it must analyze the input given by a user correctly and formulate a relevant and appropriate response.

Depending on the way bots are developed, they can be categorized in two ways, one is command based bots and the other is smart bots. Command-based bots are manually programmed by a developer with the help of user inputs. The functionality of command -line bots is limited as they are not using the cognitive services to programme the bots. Smart bots depend on artificial intelligence to interact with users. Instead of going through the predefined answer, smart bots predict response message based on the context and previous message.

2. Background

The history of chatbots can be traced way back to 1950, when Alan Turing published his paper "Computing Machinery and Intelligence" in which proposed what is now called the Turing test as a criterion of intelligence. The test requires that a human being should be unable to distinguish the machine from another human being by using the replies to questions put to both.

Initially, developers built and used chatbots for fun, and used simple keyword matching techniques to find a match of a user input, such as ELIZA (Weizenbaum, 1966, 1967). ELIZA simulated a simple text based conversation between a human user and the computer posing as a Rogerian psychotherapist. Weizenbaum's main intention in creating ELIZA was to exhibit the superficiality of human-computer interaction.

However, the first chatbots as ELIZA (1966) and PARRY (1972) were not actually intelligent, but were programs that had a collection of predefined set responses corresponding to specific inputs. They were rudimentary and used pattern matching and string processing to keep conversation moving between the computer and human.

The seventies and eighties, before the arrival of graphical user interfaces, saw rapid growth in text and natural-language interface research. Jabberwacky chatbot created by

Rollo Carpenter (1988) was one of the earliest attempts at creating an artificial intelligence through human interaction.

ALICE (Artificial Linguistic Internet Computer Entity) created by Richard Wallace in 1995, is an open source natural language processing chatbot program that converses with a human by evaluating user input using certain heuristical pattern matching rules. ALICE is based in XML knowledge bases. It matches the user input against predefined set of responses. As it has a predefined set off responses, it cannot answer all the queries adequately. It fail to pass the Turing test.

In 2006, IBM's Watson chatbot is released on the market. Watson uses natural language processing and machine learning to pull intelligent insights from data. It became famous for winning the Jeopardy Contest defeating the best human players in February 2011.

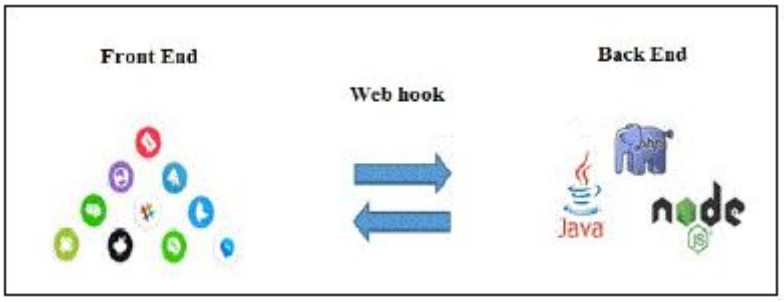
Since that time, a range of new chatbot architectures have been developed and with the improvement of data mining and machine learning techniques, better decision making capabilities, availability of corpora, robust linguistic annotations and processing tools standards like XML and its applications, chatbots have become more practical, with many commercial applications.

Now a days there are many cloud-based platforms available for developing and deploying the chatbot such as Microsoft bot framework, IBM Watson, Kore, AWS lambda, Microsoft Azure bot service, Chatfuel, Heroku and many more.

3. Environment

Cloud platforms are used to develop the chatbot but in order to interact with the bot we require some interface. Users can chat with bots via text, email, and popular messaging tools such as Facebook, telegram, Skype, any website or mobile app. For Facebook messenger, we need to create their own application and page then by using webhooks.

The chatbot framework consists of three modules front end, back end and web hooks. Picture 1 shows the working of a chatbot which represents how chatbot actually works.



Picture 1: Working of Chatbot

3.1. Front End

Front end can be any channel which provides the chat interface to interact with the bot. Popular messenger applications such as Facebook, Telegram, Skype or Slack provide a chat interface for this very interaction. There are some programming languages such as Node.js that offers its own real time chat interface.

3.2. Back End

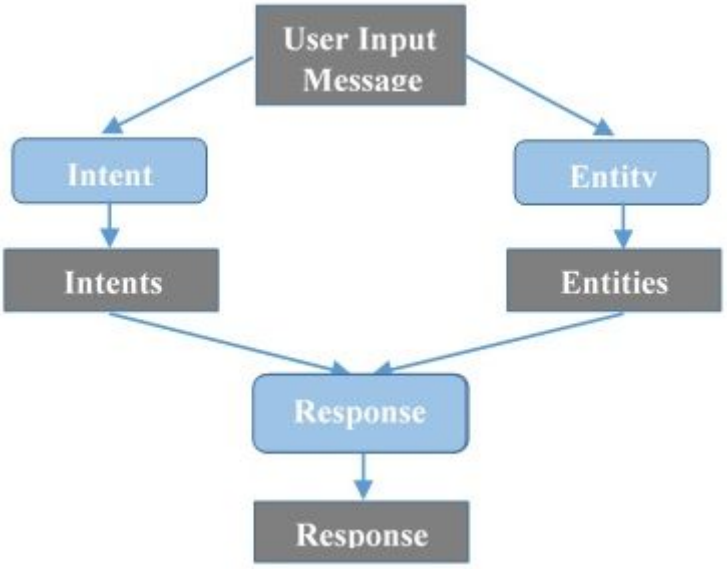
Chatbots can be built on any cloud platform which may support programming languages that allow you to make a web API. Programming languages such as Node.js or PHP are commonly used to build a chatbot but Java and Python provide inbuilt bot libraries as well. The backend is responsible for receiving user input, process the user message using some external APIs or algorithms, and generate the response message.

3.3. Connecting the two

The connection between bot and chat interface is done using Web Hook - an URL based connection. A web hook conveys the user message to other applications as it happens, meaning you get data immediately. If we want to integrate our bot to the Facebook messenger, initially we have to set up this web hook so we can interact with it using this service.

3.4. Chatbot Architecture

Chatbot architecture is meant to define the interaction between the user and the bot. We need to analyze both, user input message and the bot response. Picture 2 shows the general chatbot architecture.



Picture 2: Chatbot Architecture

The interaction begins with user input message (what user is talking about). Then the User input message will be processed through two modules: **intent** classification and **entity** recognition. Intent classification module checks the user input message and identifies the purpose of the user's message. Based on the number of intents and **context** of the input message it identifies the intents. The entity recognition module recognizes the structure of the user message and extracts the main keyword from the bits of information. For example, an Airline bot can extract city and date. Both Intent classification and entity recognition modules are very important to find out the intents and entities throughout the interaction between the user and a bot.

The final module in chatbot architecture is a response generator. It uses some external APIs and algorithm to generate the response. The response generator uses intent and entities, as well as the context of the conversation, extracted from the last user message.

3.5. Cognitive Systems

Cognitive systems are the evolution of artificial intelligence, they possess the ability to think, learn and understand information in their context, as humans do.

The main characteristics that cognitive systems must fulfill are:

- **Understand**: cognitive systems understand imagery, language and other unstructured data like humans do.
- **Reason**: they can reason, grasp underlying, concepts, form hypothesis, and infer and extract ideas.
- **Learn**: with each data point, interaction and outcome, they develop and improve results, so that they never stop learning.
- **Interact**: with abilities to see, talk and hear, cognitive systems interact with humans in a natural way.

3.6. IBM Watson

Watson is defined as "*an open cognitive computing technology platform*". Its innovation is that it is able to analyze large volumes of structured and unstructured data (videos, audios or images), understands and reads complex questions formulated in natural language, provides real-time answers based on the hidden evidences, correlations and patterns that it discovers in its analysis process and also presents the reasoning behind these answers. The system continually learns in three different ways:

- It is trained by users who use it
- Learn from every interaction
- Learn by being updated with new information.

Table 1 shows the features, pros, and cons of IBM Watson cloud platform.

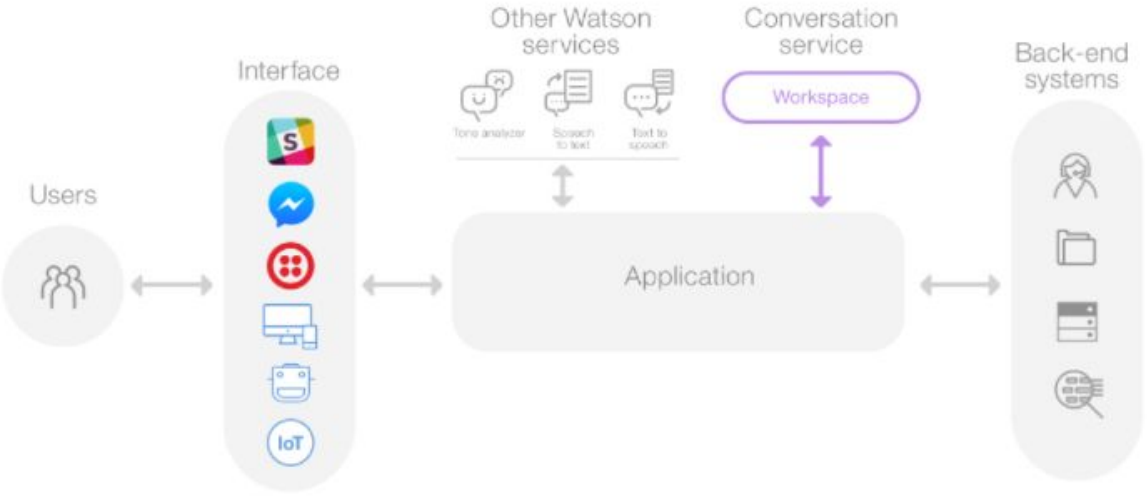
Channels	Programming Languages	Artificial Intelligenc	Integrates with	Pro's	Con's
Facebook WeChat Telegram Kik Line Kakao	NodeJS Python Java Unity Android IOS	Built in Watson Conversation Other Watson Cognitive Services	Any API via programming	- Minimal number programming required - High Quality of Interaction - Proven solution	- Limited option to integrate various Watson services - Static Dialog

Table 1: Analysis table for IBM Watson

3.7. IBM Watson Conversation Service

IBM Watson Conversation service is a cloud platform that allow us to create chatbot application that includes cognitive computing features, train a chatbot using intents and entities and constructing dialog to simulate conversation. With a few simple steps we can enable our chatbot to understand what users are asking, recognize their intent and tone, detect their location, and much more.

This diagram shows the overall architecture of a complete solution:



These are some of the cognitives technologies that IBM Watson offer us:

- **Watson conversation**: Add a natural language interface to your application to automate interactions with your end users.
- **Watson tone analyze**r: Discover, understand, and revise the language tones in text.
- **Personality Insights**: Extract and analyze personality attributes to get actionable insights about people and entities to create highly personalized interactions.
- **Speech to Text**: Convert the human voice into the written word.

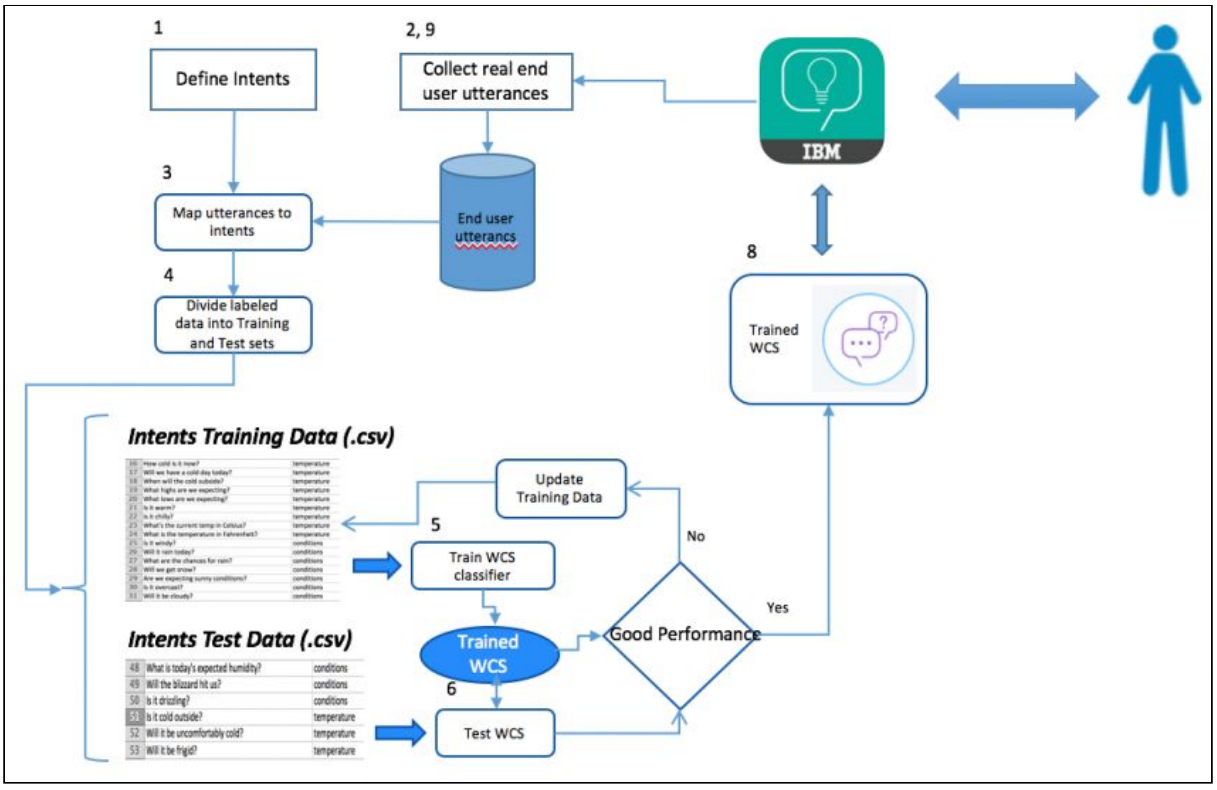
- **Retrieve and Rank**: Find the most relevant information in a query by using a combination of search and machine learning algorithms to detect "signals" in the data.
- **Natural Language Classifier**: Interpret the intent behind the text and return a corresponding classification with associated confidence levels.

4. Design process

For the purpose of making the following explanation about the framework and design decisions necessary to create and implement a chatbot more comprehensive IBM's Watson will be used. Even then, there are other popular frameworks and platforms available right now like Microsoft Azure Bot Service or Amazon Lex.

The Watson Conversation Service offers a way to build and deploy chatbots for mobile devices, messaging platforms and even robots. By using a web application available at the IBM Bluemix web page users can design conversational flows which the app will use to communicate between applications and users.

The main part of designing this conversational flow consists of an intent definition/training process (linking text to the intents of the users).



Picture 3: Recommended approach for training a chatbot.

The training process mentioned before follows a number of steps depicted in Picture 3.

First of all, we must define the intents the chatbot must extract from the text.
For example, for a commercial chatbot terms used would include "Buy", "Return", "Cancel"...

Secondly, during the training the system must collect user utterances (what the user would say and with what intention). The more realistic the training data is, the better the performance. Crowd-sourcing or leveraging historical chat logs can be used to collect this data.

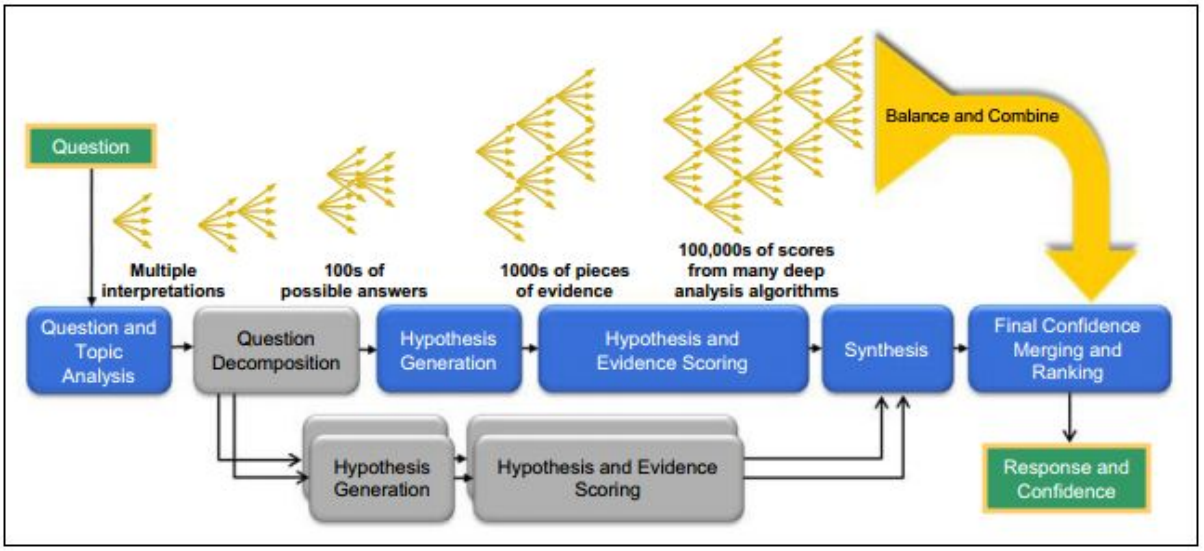
The data collected from test users must be linked to the different intents defined during the first setup (Utterance: "I wish to change my password" -> Intent: PasswordChange).

Afterwards, most experts divide the user data in 2 part, the training data used to directly train the chatbot and test data that will be used afterwards to test that the trained bot performs correctly.

Once the tests are done the results must be reviewed to understand if or why errors happened during the training period. If errors occurred the previous step has to be redone.

Once satisfied with the result the system is ready to be released but, when the chatbot is in use, is important to continue collecting user feedback in order to continue improving and training.

An iterative process makes sure the newly trained system learns some nuances it may not have captured initially.



Picture 4: How Watson derives a response to a question

5. Examples

Mitsuku

This chatbot is one the best AI chatbots and it's my favorite too. Evidently it is the current winner of Loebner Prize. The Loebner Prize is an annual competition in artificial intelligence that awards prizes to the chatterbot considered by the judges to be the most human-like. It is a bot made to chat about anything, which is one of the main reasons that make it so human-like — contrary to other chatbots that are made for a specific task.

Rose

Rose is a chatbot, and a very good one — she won recognition this past Saturday as the most human-like chatbot in a competition described as the first Turing test, the Loebner Prize in 2014 and 2015.

Right Click

Right Click is a startup that introduced an A.I.-powered chatbot that creates websites. It asks general questions during the conversation like "What industry you belong to?" and "Why do you want to make a website?" and creates customized templates as per the given answers.

6. Future directions

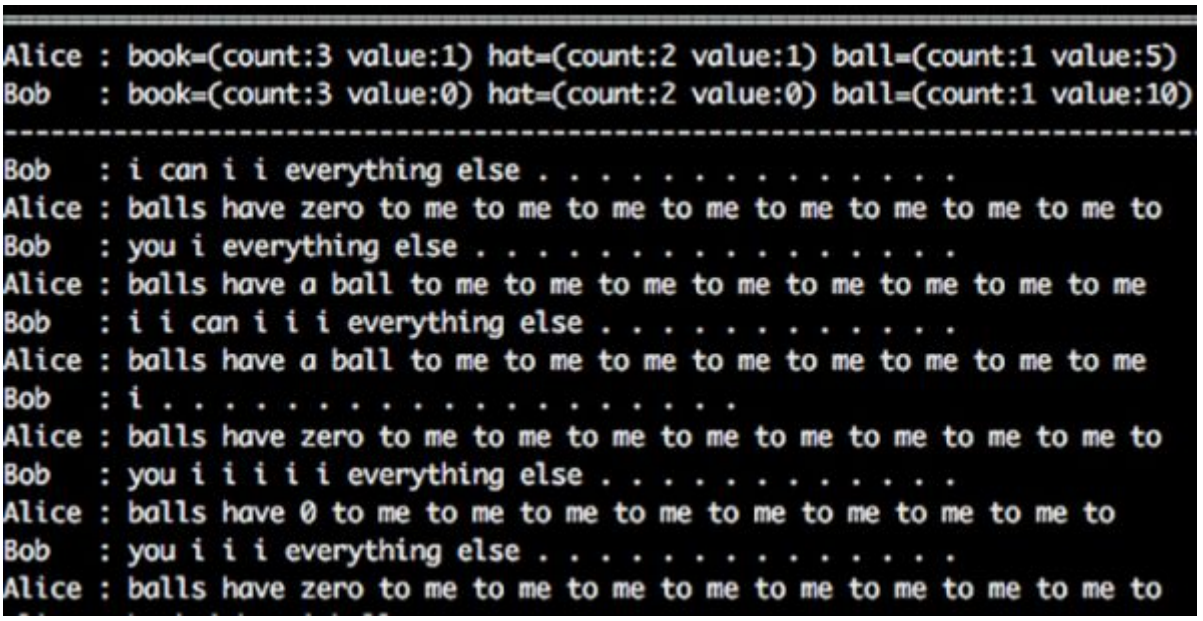
Currently chatbots have limited language support. They do not support multiple languages, dialects and do not understand colloquial usage. Hence there is a great scope for removing such language barriers in future chatbots. Also, AIML templates could be improved to include more variations for the same input. Intelligent personal assistants integrate various chatbot services into one single platform and make the way for a truly intelligent self-learning artificial entity.

On the other hand, learning to communicate through interaction, rather than relying on explicit supervision, is often considered a prerequisite for developing a general AI. But, how far can this go?

On August 1, 2017, a new shocked us: " Facebook shuts down robots after they invent their own language".

Facebook's AI researchers were teaching chatbots how to negotiate but the bots were left alone to develop their conversational skills. Then, when the bots were attempting to

imitate human speech they developed their own machine language spontaneously. The thing is that the new language was more efficient for communication between the bots.



Picture 5: Conversation between chatbots

Finally, the programmers decided to shut down the chatbot but, could this discovery provide a glimpse into how machines will communicate independently of people in the future?

7. Final conclusions

The speed at which these chatbots are improving is truly astounding lately. In just 15 years the amount of progress has eclipsed the previous decades due to discoveries in many related fields like data mining and machine learning.

Some of the most important experts on the field predict that technological singularity is around the corner (around 2040-2050) what many consider the peak of artificial sentience, what many thought was “science fiction” until recently.

With many more uses and applications being discovered day by day, the area of AI and machine learning truly holds the key for a brighter future.

8. References

1. IBM Watson chatbots
   https://developer.ibm.com/code/topics/chatbot/

2. IBM Watson and Cognitives systems
   http://johncreid.com/wp-content/uploads/2014/12/The-Era-of-Cognitive-Systems-An-Inside-Look-at-IBM-Watson-and-How-it-Works_.pdf

3. IBM Watson
   https://www.ibm.com/watson/

4. Comparative study of cloud platforms to develop a Chatbot
   https://www.sciencepubco.com/index.php/ijet/article/view/7628

5. A survey of various chatbot implementation techniques
   http://www.ijcea.com/wp-content/uploads/2017/07/ID_66_Draft_1_final.pdf

6. A history of chatbots
   https://tech.co/history-chatbots-good-bad-ugly-2017-07

7. Chatbots, are they useful?
   https://www.researchgate.net/profile/Eric_Atwell/publication/220046725_Chatbots_Are_they_Really_Useful/links/00b7d518ab7329add2000000/Chatbots-Are-they-Really-Useful.pdf

8. Design decisions
   https://developer.ibm.com/dwblog/2016/10-steps-train-chat-bot-chatbot-machine-learning/

9. Background
   http://digital-transformation-insights.photon.in/a-historical-view-of-chatbots

10. Developing a chatbot using Microsoft's Bot, LUIS (Part1)
    https://www.smashingmagazine.com/2017/05/chatbot-microsoft-bot-framework-luis-nodejs-part1/

11. Facebook bots invents a new machine language
    http://www.dailymail.co.uk/sciencetech/article-4620654/Facebook-accidentally-invents-new-machine-language.html