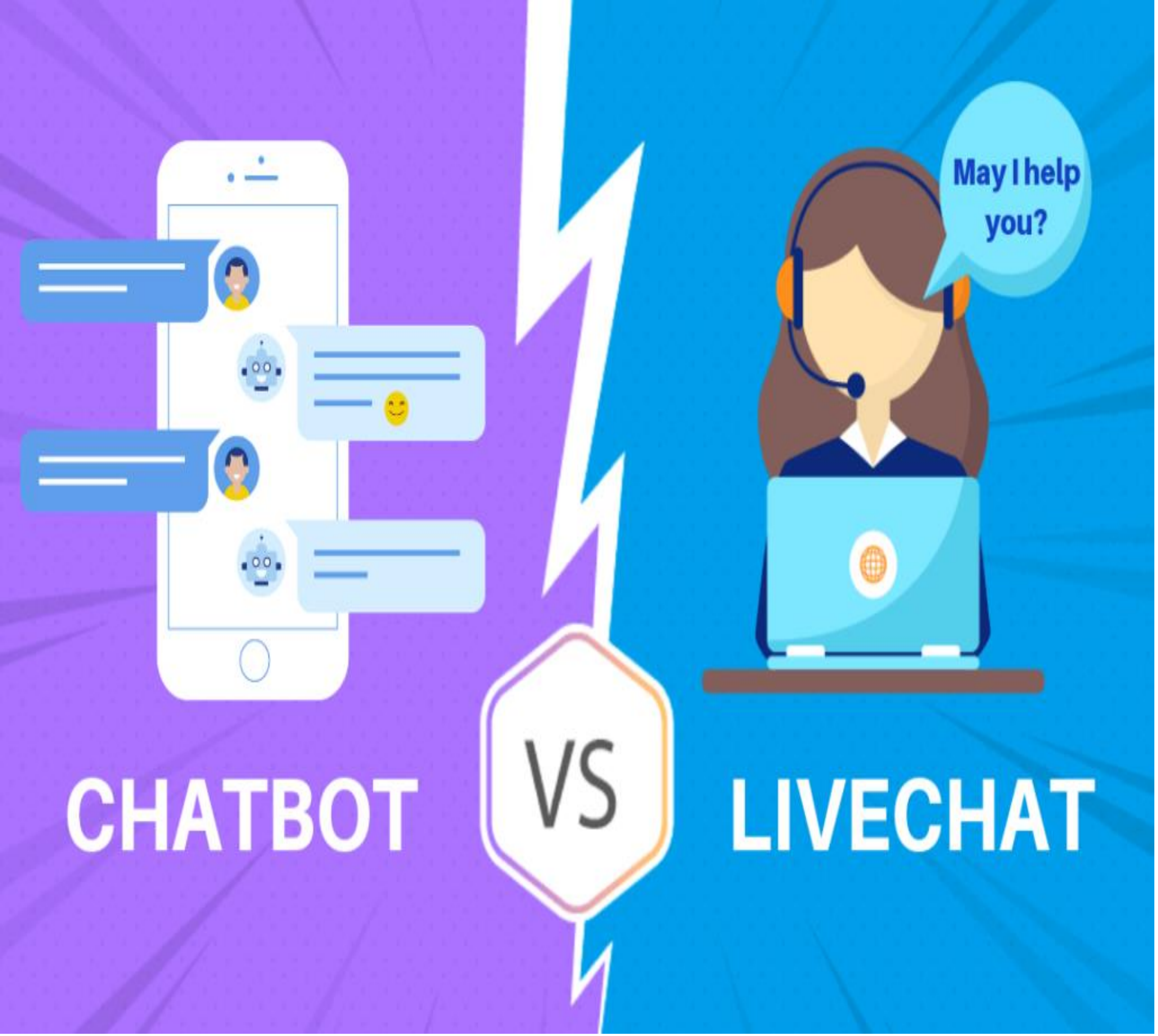


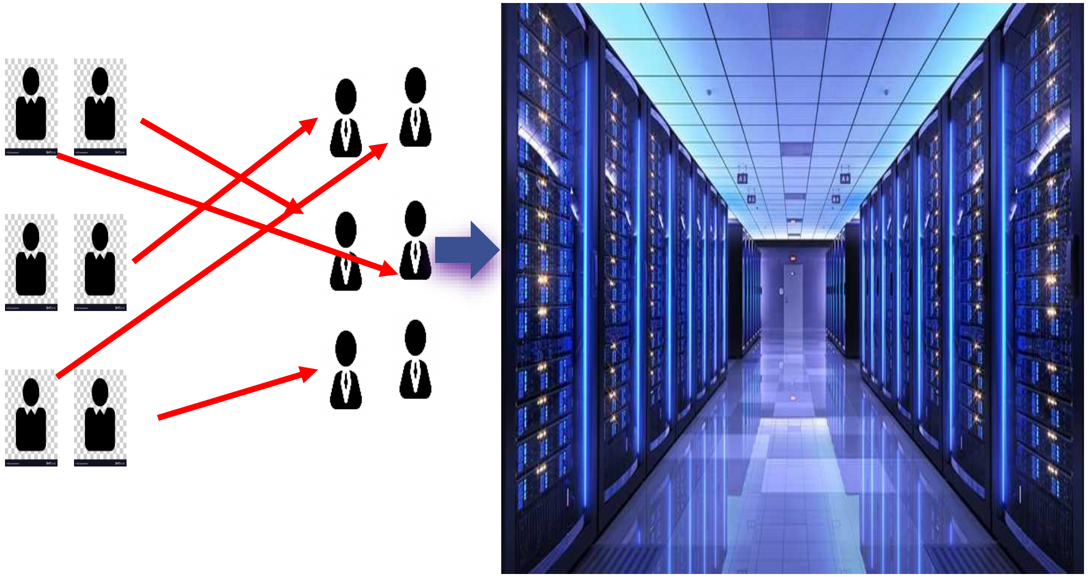
Chatbots

Agenda

- Introduction to a Chatbot
- Architecture of a Chatbot
- NLP in the cloud
- NL Interface
- How to Build a Chatbot
- Transformative user experience of chatbots
- Designing elements of a chatbot
- Best practices for chatbot development
- NLP components
- NLP wrapper to chatbots
- Audiobots and Musicbots.

Call Centre

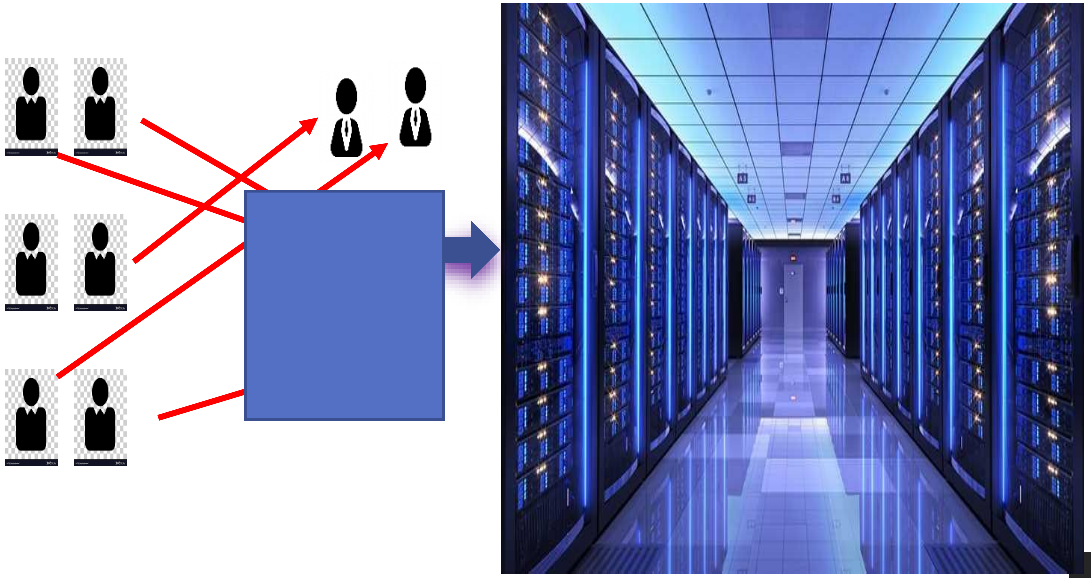
- Waiting time
- **Required huge man power**
- skill set, Investment cost
- **Infrastructure cost**

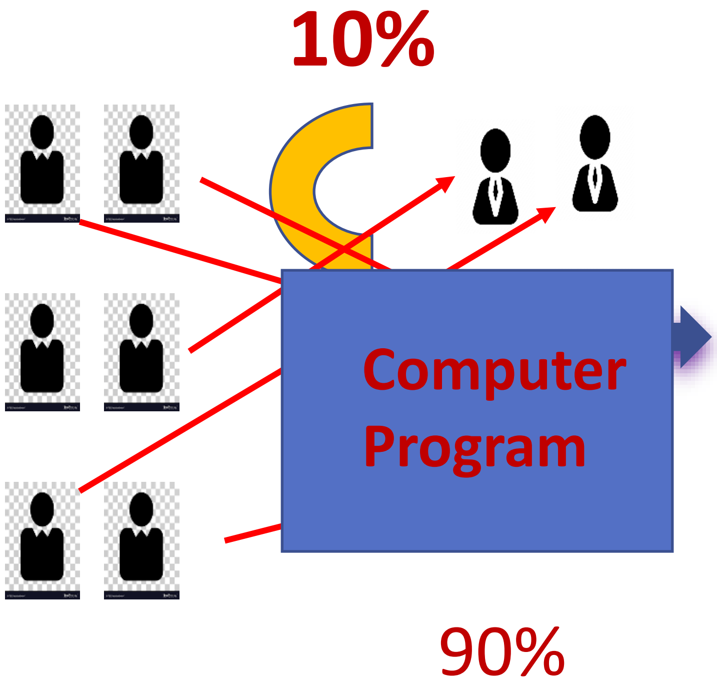


- User demand increases (it means no need wait)
- Common problems 90% cases are solved using ChatBot. For example

  Amazon

  Banking





10%

Computer Program

90%

Can take user input in natural language and have the capability to process it and generate smart and relative response.

Chat Robot

⇕

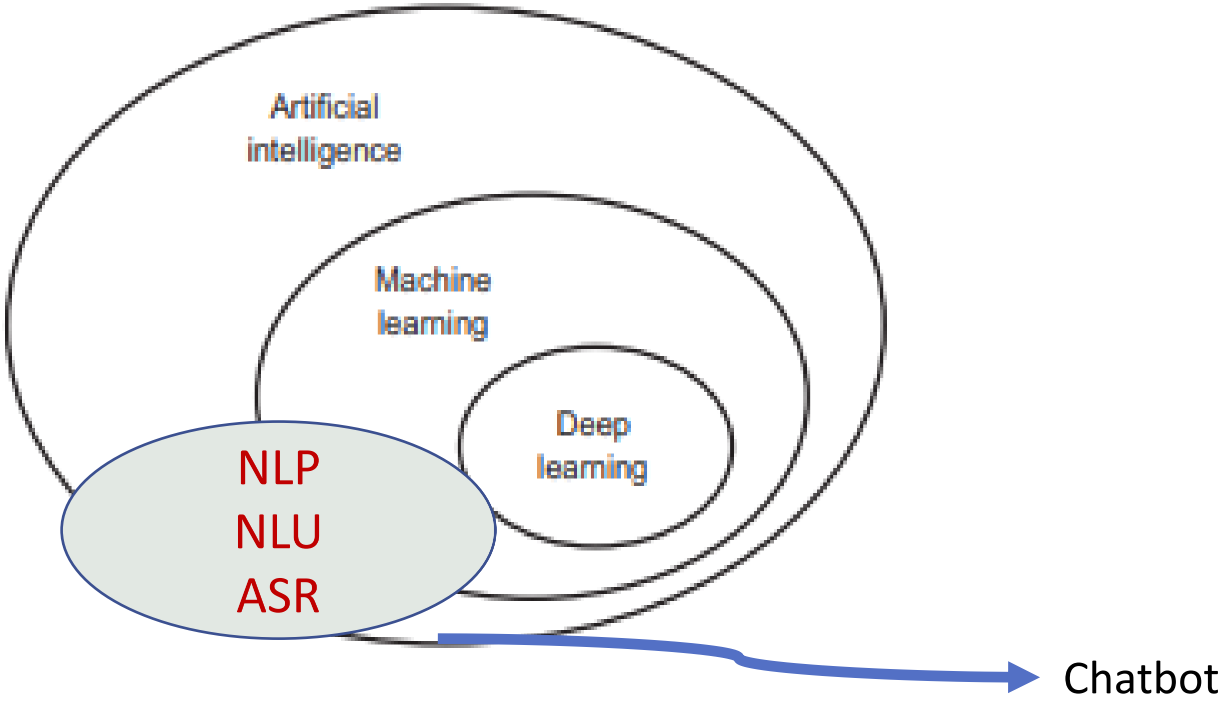
Chatbot

	Chatbot (The Chabot or technically calling - virtual assistant, is an automated program built to interact with humans via a chat interface.)	**Live Chat** (Live Chat is an online customer service system with live support of real humans sitting and answering the user's queries.)
Emotions	They don't have feelings	**Winner:** Since it is assisted by humans, trigger their emotions better
Workability	**Winner:** chatbots can work 24x7x365	it's managed by humans
Understanding	Answer pre-programmed questions	**Winner:** ability to understand the complexity of the issues.
Personalization	**Winner:** chatbot software isn't capable of storing previous data of customer's chat (Tie)	can store previous chat data
Response Time	**Winner:** Respond to querry in no time	Human being
Cost Efficiency	**Winner:** You won't have to pay it,	Hiring a team

Introduction to a Chatbot

- A **chatbot** is a AI based computer program which conducts a conversation via voice or text methods. They are also known as digital assistants that understand human capabilities.
- Chatbots are able to provide automated, intelligent responses to questions, like –
  - ✓ Queries from online customers about delivery of products
  - ✓ Providing information on the services company offer
  - ✓ Handle complaints
  - ✓ Forward users to the most appropriate human representative (i.e. to handle more complex issues)
  - ✓ Increasingly to handle direct sales.

- Chatbots are now using AI and NLP to learn from and improve upon past interactions.
- Machine learning algorithms are deployed to improve chatbot responses
  - – by analyzing data patterns of previous response
  - – while NLP is used to make the bots speak like a real human.



Let’s watch a video to understand this better!!

(https://www.youtube.com/watch?v=pX6zqaEHAdw)

Types of chatbots

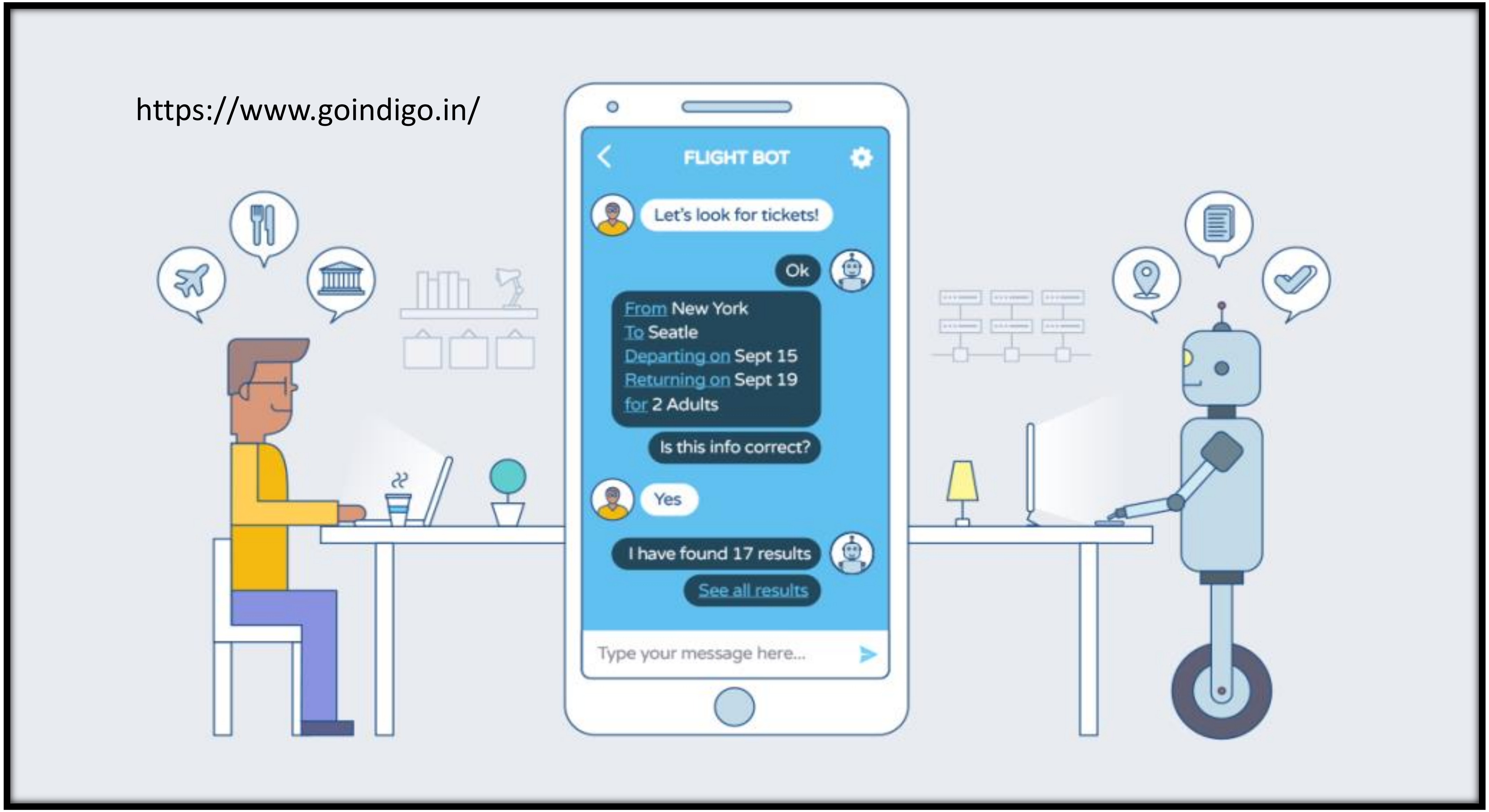
There are many types of chatbots available, a few of them can be majorly classified as follows:

•*Text-based chatbot*: In a text-based chatbot, a bot answers the user's questions via text interface.

•*Voice-based chatbot*: In a voice or speech-based chatbot, a bot answers the user's questions via a human voice interface.

Ex : Apple Siri, Amazon Alexa,.. Etc.

https://www.goindigo.in/



Let's look for tickets!

Ok

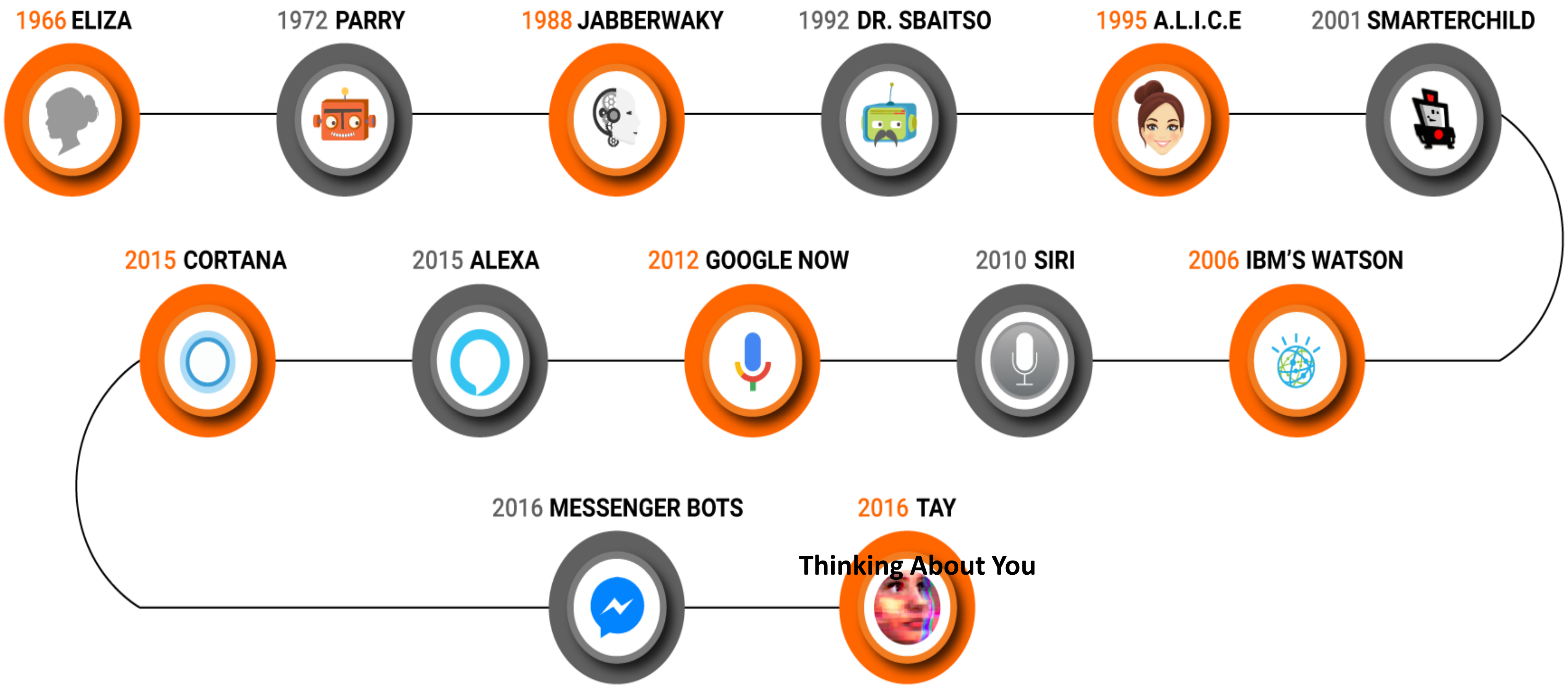
From New York
To Seatle
Departing on Sept 15
Returning on Sept 19
for 2 Adults

Is this info correct?

Yes

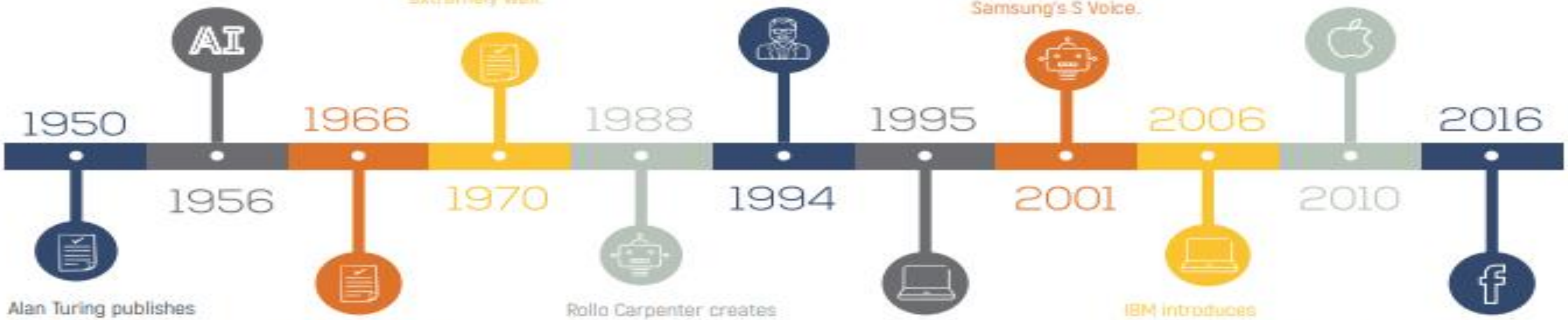
I have found 17 results

A brief history of chatbot



Thinking About You

THE HISTORY OF THE CHATBOT



1950

Alan Turing publishes Computing Machinery and Intelligence, in which he tried to determine if a machine could win what he called "The Imitation Game," resulting in the famed Turing Test.

1956

Artificial Intelligence [AI] is designated as a formal academic research discipline at The Dartmouth Conference.

1966

Joseph Weizenbaum publishes ELIZA, a program for mimicking human conversation. ELIZA works by parsing the words that users enter into a computer and then matches them to a list of possible scripted responses.

1970

Terry Winograd publishes SHRDLU, a natural language system working in restricted "blocks worlds" with restricted vocabularies that work extremely well.

1988

Rollo Carpenter creates Jabberwacky, a bot aiming to "simulate natural human chat in an interesting, entertaining, and humorous manner."

1994

Michael Mauldin coins the term "ChatterBot."

1995

Richard Wallace develops Artificial Intelligence Markup Language [AIML], forming the foundation for Artificial Linguistic Internet Computer Entity" [A.L.I.C.E.].

2001

SmarterChild bot is distributed widely across SMS and messaging platforms, featuring personalized conversations and quick data access. Considered the precursor to Apple's Siri and Samsung's S Voice.

2006

IBM introduces Watson, designed specifically to take on human contestants on the game show Jeopardy! using Natural Language Processing [NLP].

2010

Apple creates Siri, an intelligent personal assistant featuring a natural language UI to answer voice commands and perform Internet service requests.

2016

Facebook allows developers to create interactive bots on the Facebook Messenger platform. Within the first two months, more than 10,000 bots were available.

Approaches to design chatbots

There are mainly two approaches used to design the chatbots, described as follows:

- In a Rule-based approach, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex. The bots can handle simple queries but fail to manage complex ones. (Ex: What is the weather today ? Ans: Only one line)
- Self-learning/AI based bots are the ones that use some Machine Learning-based approaches and are definitely more efficient than rule-based bots. (Ex: Book one ticket from BLR to US? Ans: Different routes)
- These bots can be further classified in two types:

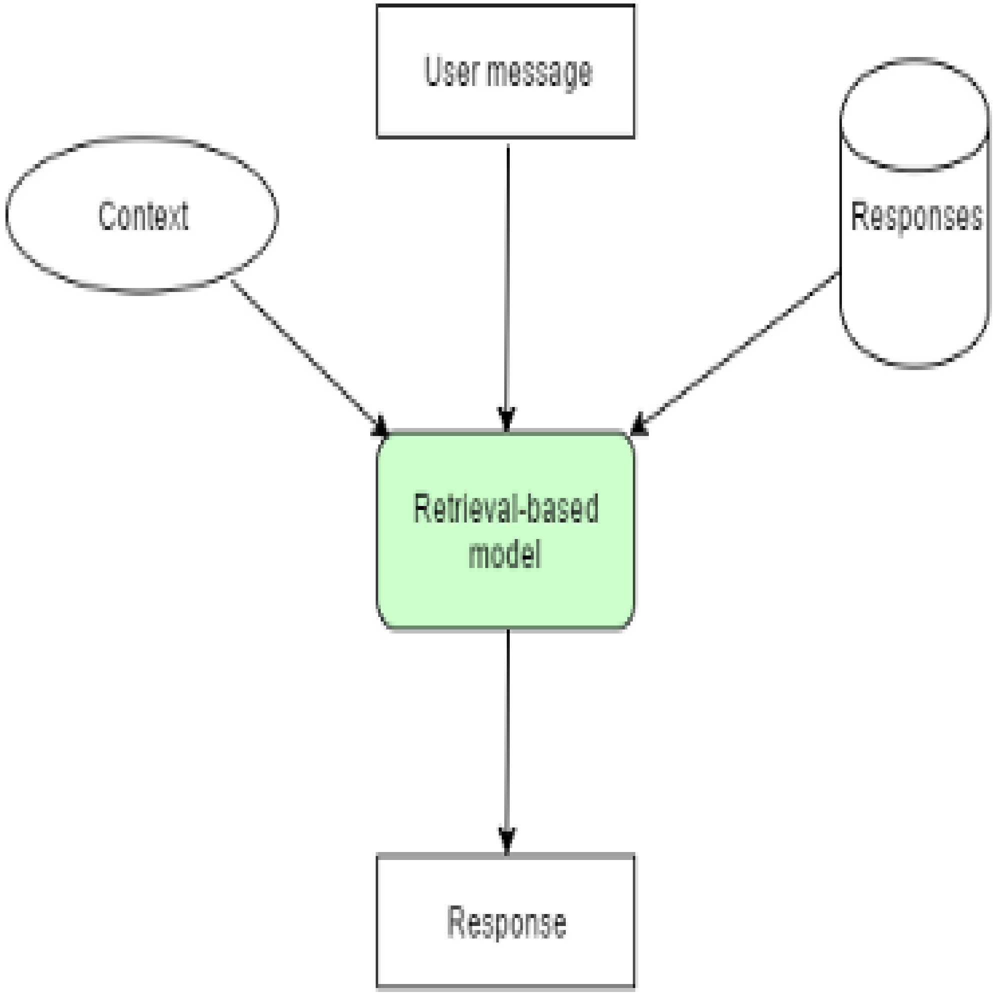
  1. Retrieval Based Modeling
  2. Generative Modeling

Retrieval Based Modelling

Retrieval-based models are more practical at the moment, many algorithms and APIs are readily available for developers.

The chatbot uses the message and context of conversation for selecting the best response from a predefined list of bot messages.

The context can include current position in the dialog tree, all previous messages in the conversation, previously saved variables (e.g. username).



How a response is selected ?

1) Pattern-based heuristics

AIML is a widely used language for writing patterns and response templates. Bot developers write code in AIML language, code can include multiple units like this:

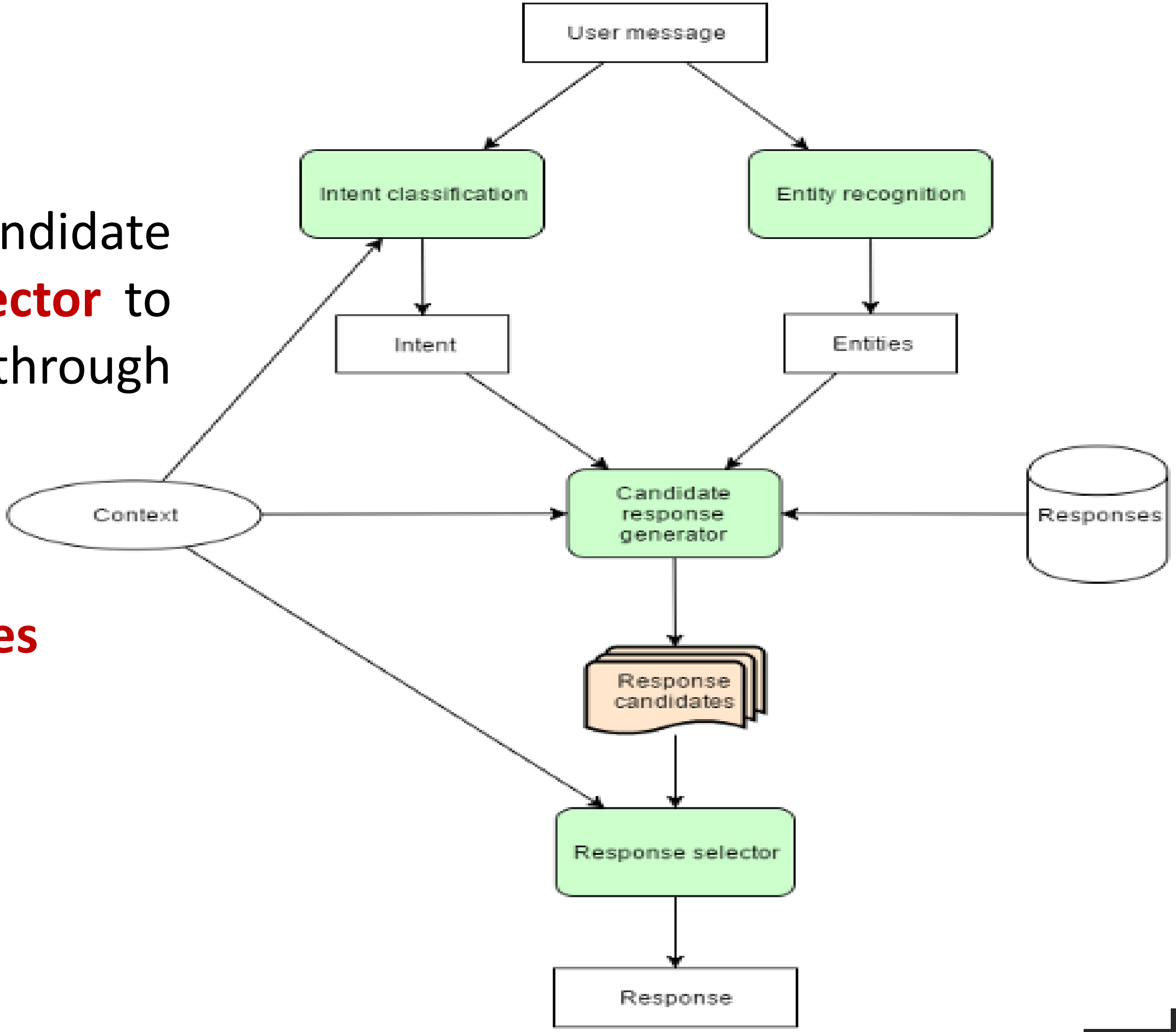
```
<category>
<pattern>WHAT IS YOUR NAME</pattern>
<template>My name is Google Assistant.</template>
</category>
```

When the chatbot receives a message, it goes through all the patterns until finds a pattern which matches user message. If the match is found, the chatbot uses the corresponding template to generate a response.

2) Architecture with response selection

An architecture of Chatbot requires a candidate **response generator** and **response selector** to give the response to the user’s queries through text, images, and voice.

use a repository of predefined responses



Working principle:

user messages are given to an intent classification and entity recognition.

Intent: An intent in the above figure is defined as a user’s intention, example the intent of the word “Good Bye” is to end the conversation similarly, the intent of the word “What are some good indian restaurants” the intent would be to find a restaurant.

Entity: An entity in the Chatbot is used to modifies an intent and there are three types of entities they are system entity, developer entity and session entity.

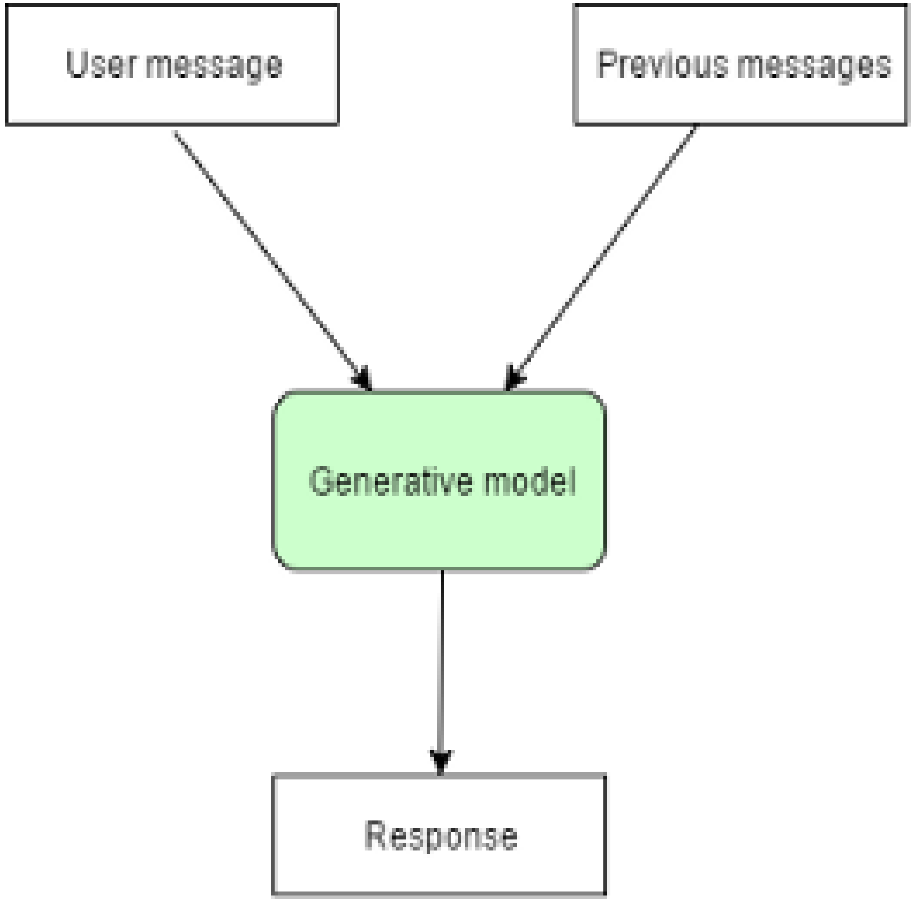
Candidate Response Generator: The candidate response generator in the Chatbot do the calculations using different algorithms to process the user request. Then the result of these calculations is the candidate’s response.

Response Selector: The response selector in the Chatbot used to select the word or text according to the user queries to give a response to the users which should work better.

Generative Modelling

Generative models are the future of chatbots, they make bots smarter. This approach is not widely used by chatbot developers.

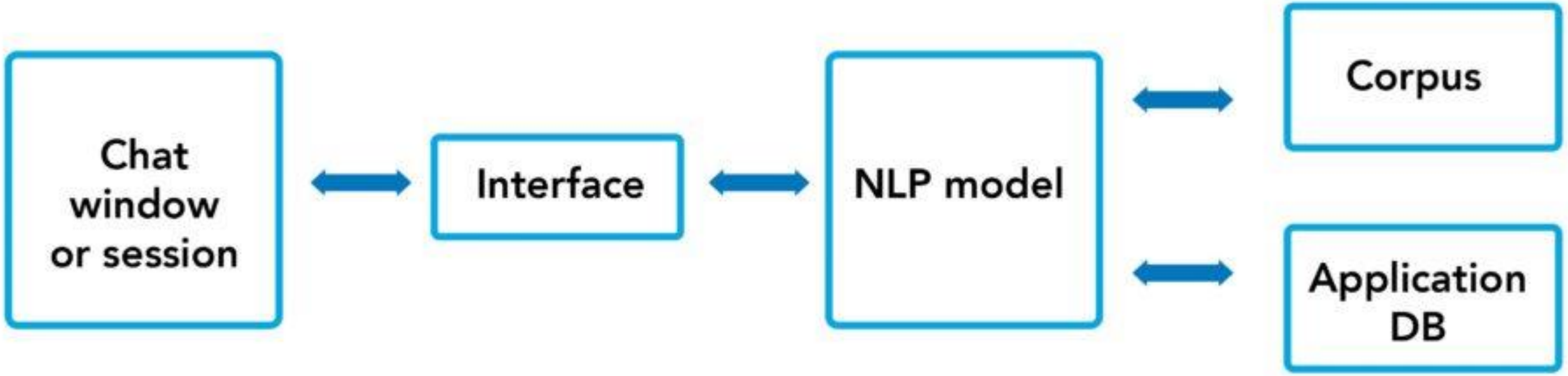
- Generative models are comparatively difficult to build and develop.
- Training of this type of bot requires investing a lot of time and effort by giving millions of examples.
- This is how the deep learning model can engage in conversation.
- **The output not only depends on the current input, but to a series of input given in the past.**



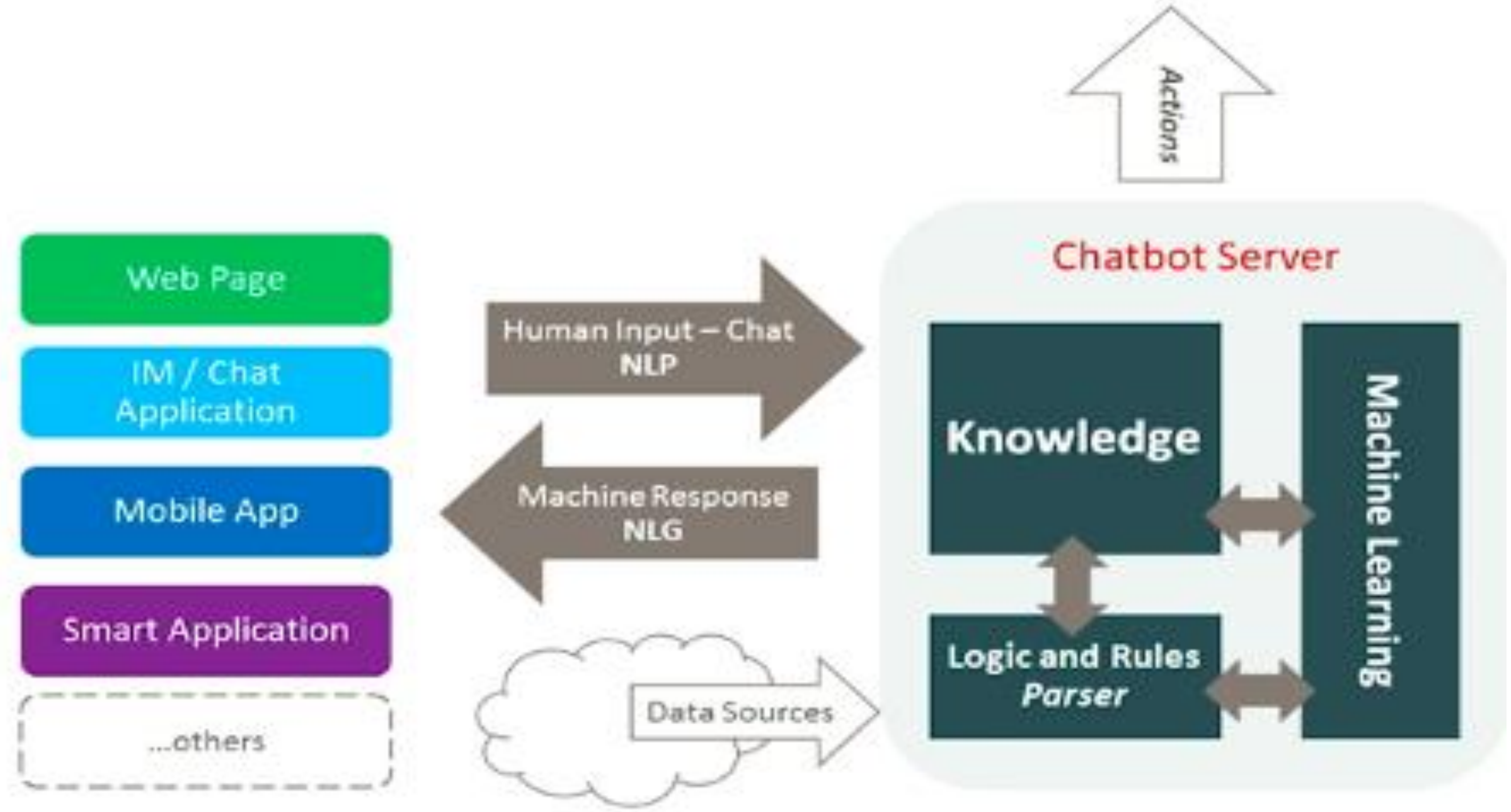
Architecture of a Chatbot

Typical chatbot architecture should consist of the following:

- Chat window/ session/ or front end application interface
- The deep learning model for Natural Language Processing [NLP]
- Corpus or training data for training the NLP model
- Application Database for processing actions to be performed by the chatbot

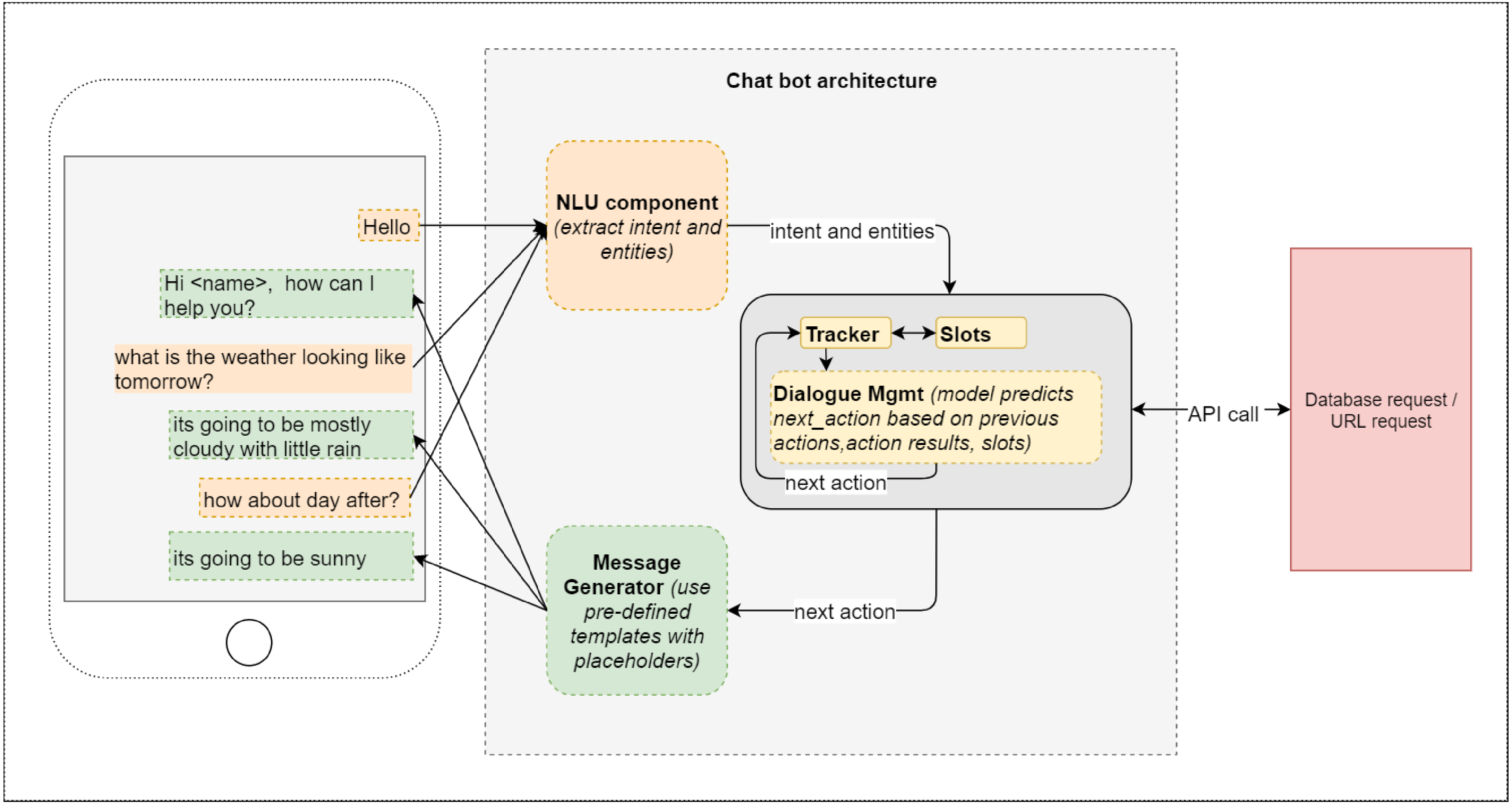


Anatomy of a Chatbot



The architecture of a chatbot depends on classes of chatbots. There are two classes of Chatbots, some driven by intents (conversation) others based on actions (services).

Conversational AI Chatbot



Hello
Hi <name>, how can I help you?
what is the weather looking like tomorrow?
its going to be mostly cloudy with little rain
how about day after?
its going to be sunny
Chat bot architecture
NLU component (extract intent and entities)
intent and entities
Tracker
Slots
Dialogue Mgmt (model predicts next_action based on previous actions,action results, slots)
next action
API call
Database request / URL request
next action
Message Generator (use pre-defined templates with placeholders)

NLU component (*Natural Language Understanding*) helps in extracting the intent and entities from the user request.

Intent : Intent capture the user goal.
what the user asking for?

Hello
hi
good morning

#greetings

A supervised intent classification model that is trained on varieties of sentences as input and intents as target. Typically, a linear SVM will be enough as an intent classification model.

Ex : user : Hello
bot : detect #greetings
user : when your store opens?/when do you close?
bot : #hours_info

Entity: Capture specific information in the user input.

Ex: date and time – when , location – where, number – how many

user : when is miyapur central opens?
bot : detect #hours_info & @location : miyapur

Rule to specify entity :

Entity value should not contain spaces.

@location : New York ✗
@location : NewYork ✓

A Entity extraction model — This can be a pre-trained model like Spacy or Stanford NLP library (OR) it can be trained using some probabilistic models like CRF (conditional random fields).

Dialogue Management

The dialog determines the response based on the detected intents and/or entities in input.

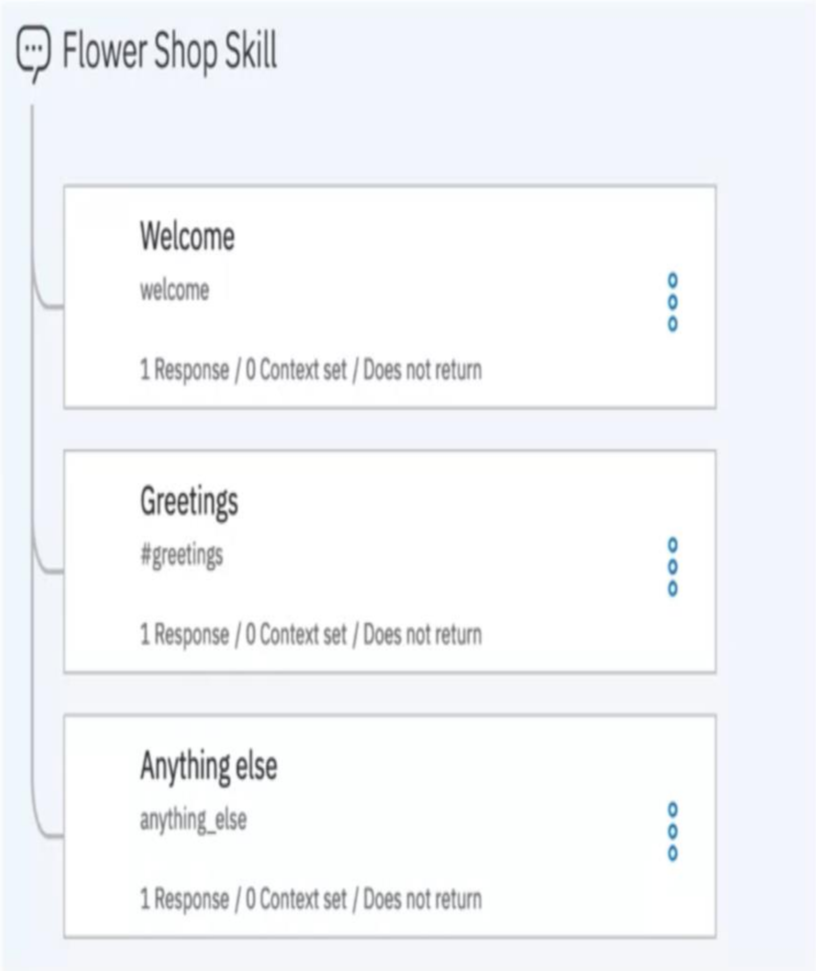
Node Name – condition – Response – next action

This model should be context aware and look back into the conversational history to predict the next_action.

This is similar to a time-series model and hence can be best captured in the memory state of the LSTM (Long Short-Term Memory) model.

Now, the predicted value of the next_action can be something like —

- Respond to the user with an appropriate message
- Retrieve some data from a database (if have any)
- Make an API call and get some results matching the intent.

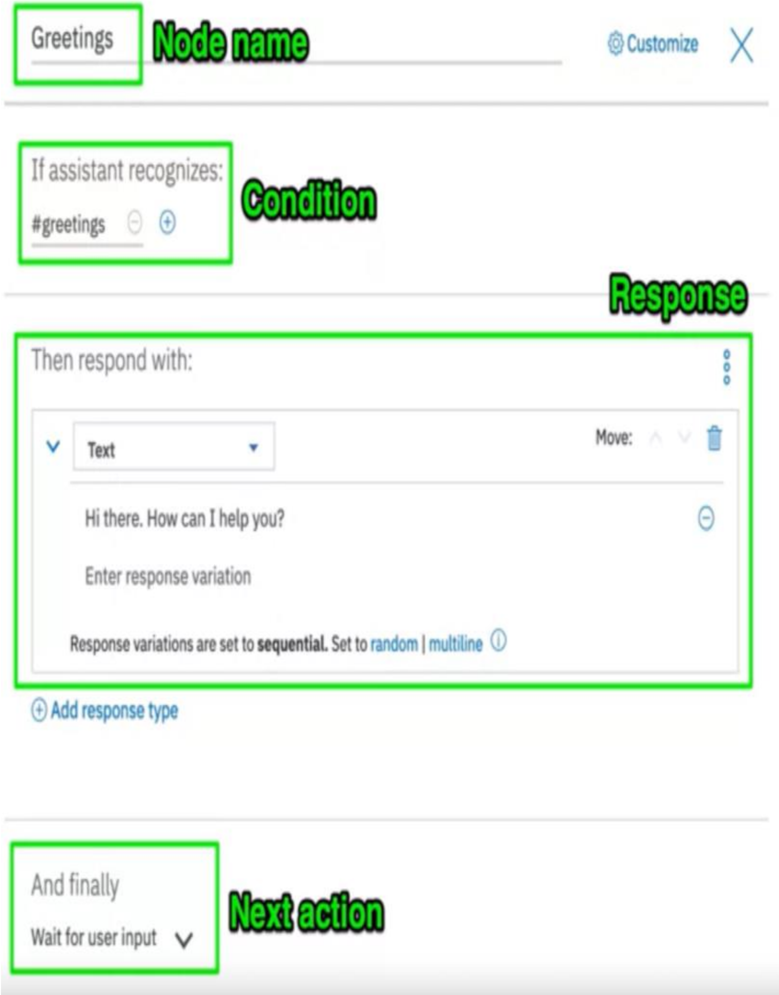


Flower Shop Skill

Welcome
welcome
1 Response / 0 Context set / Does not return

Greetings
#greetings
1 Response / 0 Context set / Does not return

Anything else
anything_else
1 Response / 0 Context set / Does not return



Greetings — Node name

⚙ Customize X

If assistant recognizes:
#greetings ⊖ ⊕

Condition

Response

Then respond with:

Text ▾ — Move: ∧ ∨

Hi there. How can I help you?

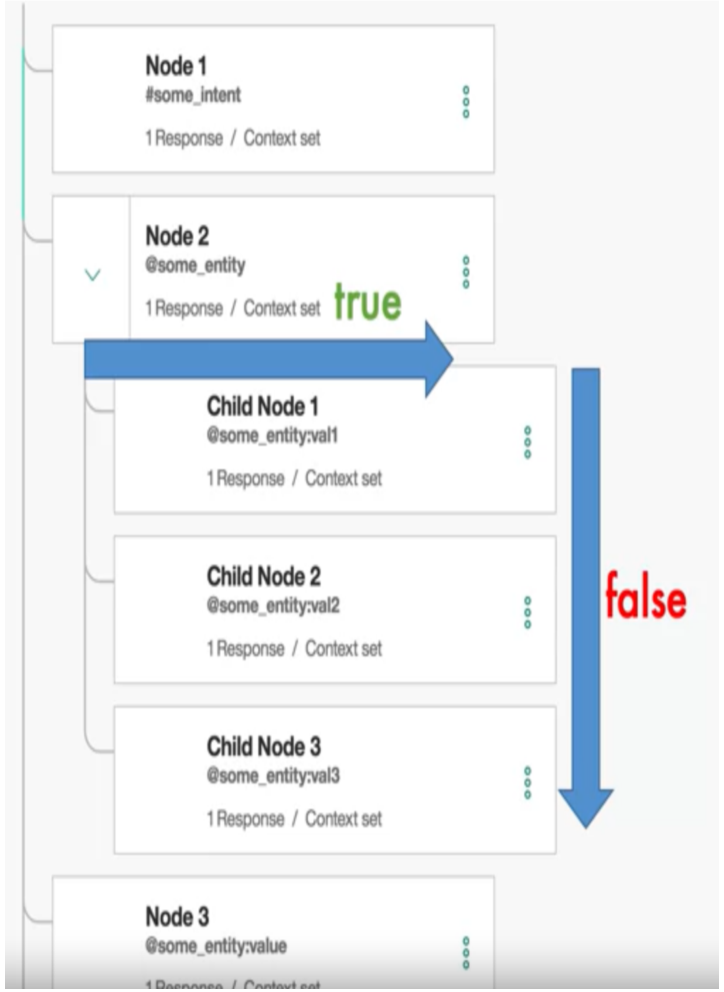
Enter response variation

Response variations are set to **sequential.** Set to random | multiline ⓘ

⊕ Add response type

And finally
Wait for user input ∨

Next action



Node 1
#some_intent
1 Response / Context set

Node 2
@some_entity
1 Response / Context set

true

Child Node 1
@some_entity:val1
1 Response / Context set

Child Node 2
@some_entity:val2
1 Response / Context set

false

Child Node 3
@some_entity:val3
1 Response / Context set

Node 3
@some_entity:value
1 Response / Context set

MESSAGE GENERATOR

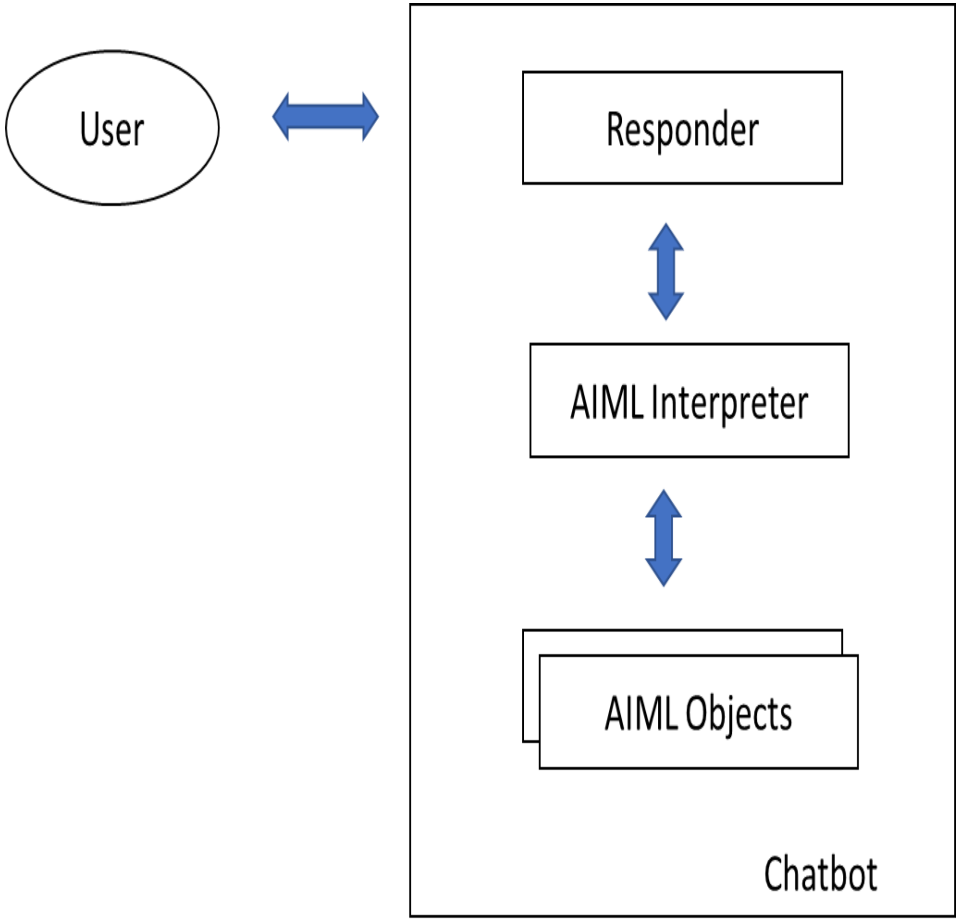
- Message generator component consists of several user defined templates (templates are nothing but sentences with some placeholders, as appropriate) that map to the action names.

- Depending on the action predicted by the dialogue manager, the respective template message is invoked.

Action based AI chatbot

- Action-based AI chat-bot are built using Rule-Based Approach. In a rule-based approach, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex.

- The bots can handle simple queries but fail to manage complex queries. Hence, the bot can never pass the Turing test if based on some rule-based models.



How to Build a Chatbot

A chatbot can be build in two ways

1. Using a framework (By using online tools)
2. From Scratch(by using Programming)

Every chatbot will contain front end and back end

Front End : The Messaging Channel

- Messaging channel acts as a user interface (through which user communicates)
- Can be leverage on existing messaging channels such as slack, Facebook, etc.
- can also build your own messaging layer such as a custom website or mobile app

Back End : NLP Technology – Text classifiers

Here’s a more specific example. When talking to a customer service agent, a customer might say any of the following:

- “How come I can’t log into my account anymore?”
- “I forgot my password.”
- “It says my password is incorrect.”
- “I’m locked out of my account.”

- Luckily, all these requests have the same solution, which is to reset the customer password.
- So, the core functionality of a chatbot is to map all possible user inputs into a much smaller set of responses. This type of pattern recognition is what text classifiers do best.

Anatomy of a chatbot backend

- A framework, the backend of your chatbot will most likely consist of three main parts: intents, entities, and dialog.
- Intent – An intent is the purpose of a user’s input.
- Entity –Entities are the nouns.

They ‘re the keywords in a user’s input.

If a particular word differentiates one input from another, it probably should be an entity.

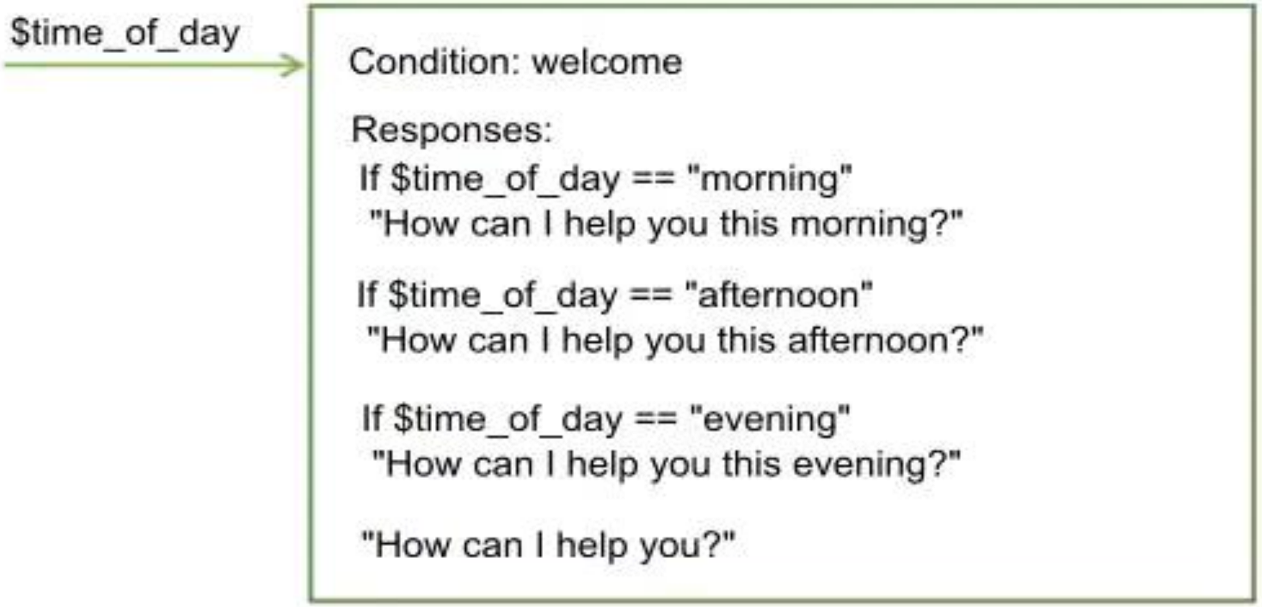
For example if a user wants to find the business hours of a bank’s branch, the location of the bank would be one of the entities.

- Dialog - The dialog determines the response based on the detected intents and/or entities in input.

- Context Variable:

A context variable is a variable that you define in a node.

A context variable contains the information shared between the framework and your application.



$time_of_day

Condition: welcome

Responses:
If $time_of_day == "morning"
"How can I help you this morning?"

If $time_of_day == "afternoon"
"How can I help you this afternoon?"

If $time_of_day == "evening"
"How can I help you this evening?"

"How can I help you?"

- Human in the loop
- Embedding a human into your chatbot has two significant benefits.
- If the chatbot is not confident about its response, it can ask the human for approvals or edits before sending it to the end user.
- Another setup is to have the chatbot provide multiple responses, and have the human choose the most appropriate one.
- The second benefit is that humans are the best way to prevent your chatbot from utterly failing your users. There are a few ways to detect if the conversation needs to be routed to a human:
  - The chatbot doesn’t understand the user’s input—this usually means the user input doesn’t match any of your established intents.
  - The conversation is taking too long, or a circular pattern is detected.
  - Negative sentiment is caught in the user’s input.
  - The user directly asks to talk to a real person.

Transformative user experience of chatbots

- Customer care strategy by next three years, while 62% will probably achieve this goal by next year itself. In a world where people are already accustomed to Siri, Alexa and Cortana, there will always be more room for entities like Nina, the chatbot from Bank of Sweden or its counterpart, Erica, from Bank of America.

How the implementation of chatbots is redefining customer experience?

- Catering to the Culture of Immediacy(24/7 available)
- Providing a Highly Targeted User Experience (website is not good enough/chatbot)
  - Chatbot uses customer data to improve its conversational patterns and learn more about user preferences.
- Humanizing Brand to Build Stronger Relationships :
  - Inject personality into chatbot language – reflect brands tone and voice
- Inspiring Natural, Human-Like Conversations.

Uses of Chatbots

- 24/7 availability
- 100% personalization
- Natural language processing (NLP)
- Efficiency & costs

Designing elements of a chatbot

- Completely scripted, rules based bot
- 100% machine learning, AI-based bot
- Some hybrid of experience 1 and 2

Chatbot Design Elements:

- Buttons
- Cards
- carousel
- Postback
- Quick Reply
- Smart Reply
- Persistent Menu

Chatbot Design Principles

1. Find a problem or a user need and discover whether a new chatbot design will solve it.
2. Chatbot Design Flows
3. Remember
4. Get specific and limit your offerings
5. Make sure the users know they’re talking to a bot
6. On boarding / Greeting
7. Buttons for quick answers
8. Media elements
9. UI

Best practices for chatbot development

- Know your customer
- Identifying correct use case scenarios
- Choosing the right Bot Development Framework
- Connecting the relevant systems
- Lucid Conversation
- Expectation of Failure
- Introduce Your Chatbot to First-Time Users
- Add Variations to Your Responses
- Make a Main Menu That’s Accessible Anywhere
- Have Context Awareness
- Be Able to Fix Incorrect Inputs
- Handle the “I Do Not Understand” case
- Be Careful About Creating a Personality

NLP components/NLP wrapper to chatbots

- Natural Language Processing (NLP)
  - Natural Language Understanding (NLU) – Analyzing Text
  - Natural Language Generation(NLG) – Text Generation

Fundamental Methods of NLP for Building Chatbots

- POS Tagging
- Stemming and Lemmatization
- Named-Entity Recognition
- Stop Words
- Dependency Parsing
- Noun Chunks
- Finding Similarity
- Tokenization
- Regular Expressions

Reasons for the rise of chatbots

- Availability of NLP capabilities in the cloud
- The Proliferation of messaging platforms
- The push for Natural Language Interfaces

NLP in the cloud

- The availability of NLP capabilities in the cloud has been the most potent force behind the rise of chatbots.
- NLP, Text classifiers and Entity extractors– Core functionalities of Chatbots
- With open source solutions and cloud API’s – a much superior user experience is possible.
- It enables many more enterprises to use this technology

Google Cloud Platform
aws
Microsoft
Azure

NL Interface

- In most human-machine interactions, users translate their intentions into a series of keystrokes and button clicks.
- Then the machines responds via pixels on a screen.
- The Proliferation of devices such as amazon echo has drawn developers toward the idea of a voice controlled home.
- It would be nice to talk to computer the way we talk to each other.

Audio bots / Music bots

- Music bots can find songs on YouTube, Spotify, or other platforms and play them in a Discord voice channel. If you join the voice channel, you can listen along.
- This provides a great way to share new music with friends, host listening parties, or provide background music
- The Best Discord Bots: (Online)
  - FredBoat
  - Groovy
  - Rythm

Save Trees

And

Save Power