# Chatbot survey

# Concept of Chatbot:

# Problem faces the companies before Chatbots:

Customer satisfaction with a company’s services is often seen as the key to success and long-term competitiveness for a company

For example ,as customers sometime cannot understand how to use a specific product in a such company so they direct to call customer service for asking about what they want to know , so the employees in customer service (specifically in call centers) had to sift through long documents to find the answer.

As a result, the only way to get help quickly was to pick up the phone and talk to underwriting or sales support – even for answers to FAQs or to basic “how-to” questions. This overloaded the call centres, resulting in long wait times as it takes a long time to process a single request.

As a result, customer experience their interactions disappointed and dissatisfied which reduces the throughput and business performance drastically. Research showed that nearly 75% of customers have experienced poor customer service.

# Solution :

In order to truly be effective and make business processes automated an alternate system is required. An advance dialogue system known as AI chatbot application system could automate the entire business processes.

# So ,What is the chatbot ?

A chatbot is a computer program which conduct conversation in natural language via speech or text , understands what the user is trying to say , and reply with relevant message based on business rules and data of organization.

And also chatbot application system must have natural language processing (NLP), deep neural networks (DNN) so that it can understand what customers are looking for and can analyse data better than humans .

# An overview about the history of chatbot:

This technology started in the 1960’s; the aim was to see if chatbot systems could fool users that they were real humans.

# The example of traditional & old Chatbots :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chatbot | Year | Description | Open source/ proprietary | Technology/  Approach | Self-learning | Type |
| Eliza | 1966 | Simulation of psychotherapist,  rephrasing her response with few grammar rules | Open source | Basic Pattern matching | NO | Text |
| Parry | 1972 | PARRY attempt to simulate the behavior of person with paranoid schizophrenia | proprietary | Like Eliza chatbot | NO | Text |
| Jabberwacky | 1982 | This Chatbot aim to simulate natural human chat in an interesting, entertaining and humorous manners | proprietary | Contextual pattern matching ,learns by association ,storing replies to inputs in a database | Yes | Text |
| Alice | 1995 | A natural language processing chatbot-a program that engages in a conversation with human by applying some heuristic pattern matching | Open source | AIML | NO | Text |

# The Drawbacks of this traditional and old chatbots:

* No self-learning.
* The traditional chatbot’s dialogue capability is too inflexible. It can answer to the user only if there is a pattern (lexical) matching between the user query and set of question-answer stored in its knowledge base . The answers are given using a set of predefined responses.
* Traditional chatbots are lacking in the intuitive capability of human beings to see the meaning, relationships and possibilities beyond the reach of sense.

As result , this traditional chatbots are not effective and customers are not satisfied , as there is no sense between human and computer . So nowadays , developers make an innovation by using DL(deep learning) in the chatbot applications.

There are two main tasks in deep learning (DL):

* The first is to extract meaning from the input.
* The second is to generate an output from that, either a translation or a response in the case of a chatbot application.

Challenges in developing a good model:

The major challenge is that creates an adequate sense of context and effectively related inputs to outputs. The sequence-to-sequence (seq2seq) model in deep recurrent neural networks (DNN) with attention mechanism provides an appropriate architecture to meet these challenges.

# Chatbot applications are classified into:

* Chatbot applications can be grouped into four different categories, namely **service**, **commercial**, **entertainment** and **advisory chatbot** .

Service chatbots : are designed to provide facilities to customers. For example, logistics firm to respond to questions about deliveries and provide copies of dispatch documents through instant messaging channel rather than emails or phone calls.

Commercial chatbots: are designed to streamline purchases for customers. For example, a pizza company can take delivery orders or notify promotions via messaging interface.

Entertainment chatbots : are designed to keep customers engaged with sports, favourite band, movies or other events. It offers the option of placing bets, detail on upcoming events and ticket deals.

Advisory chatbots: are designed to provide suggestions, give recommendations on service, offer maintenance or repair goods. This type of chatbot can contact people, offer support and advice tips when it is needed.

* chatbot application can be classified into two groups such as **task-oriented** and **non-task-oriented**.

Task-oriented chatbots: aim to assist the customers to complete certain tasks and have short conversations. For example, Siri, Google Now, Alexa dialogue agents can give travel directions, find restaurants and help to make phone calls or texts.

Non-task-oriented chatbots: focus on conversing with customers to answer questions and entertainment.

* chatbot applications can be also dividedinto four groups such as **goal-based**, **knowledge-based**, **service-based** and  **response generated-based.**

Goal-based chatbots: are classified based on the primary goal aim to achieve. They are designed for particular task and setup to have short conversations to get information from the user to complete the task. For example, a company deploys chatbot on their websites to help the customer to answer their question or address problems.

Service-based chatbots: are classified based on facilities provides to the customer. It could be personal or commercial purpose. For example, logistics company could provide copies of dispatch documents through chatbot rather than phone calls or customer can make a meal order from MacDonald.

Knowledge-based chatbots: are classified based on the knowledge they access from the underlying data sources or the amount of data they are trained on.The two main data sources are open-domain and closed-domain. Open-domain data sources answer depends on general topics and respond appropriately. Closed-domain data sources focus on a particular knowledge domain. All information required for answering the question is provided in the dataset itself such as Daily Mail.

Response Generated-based chatbots: are classified based on what action they perform in response generation. The response models take input and output in natural language text. The dialogue manager is responsible for combining response models together. To generate a response, dialogue manager follows three steps. First, it uses all response models to generate a set of responses. Second, returns a response based on priority. Third, if no priority response, the response is selected by the model selection policy.

# NOW , Focus on Response Generated-Based Chatbot as it is related to our Chatbot project.

there are various response models that are based on four categories namely—**Template-based Model** ,**Generative Model**, **Retrieval-based Model** and **Search Engine ModeI**.

## Template- based Model:

This Model uses AIML and Pattern Matching: Artificial Intelligence Markup Language (AIML) is most popularly used for writing patterns and response in the process of chatbot development.

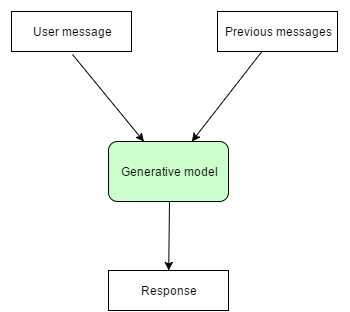
Example of AIML:



With a natural language processing pipeline and predefined rich pattern, AIML can be used to build a smart chatbot. These bots parse user message, find synonyms and concepts, tag parts of speech and find out which rule matches the user query. However, these bots do not run machine learning algorithms or any other APIs unless specially programmed.

## Generative-Model:

This model is used for the development of smart bots that are quite advanced in nature. This type of chatbot is very rarely used, as it requires the implementation of complex algorithms.



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. However, still, we cannot be sure what responses the model will generate. where they are able to generate new messages based on the context and/or current states of the conversation

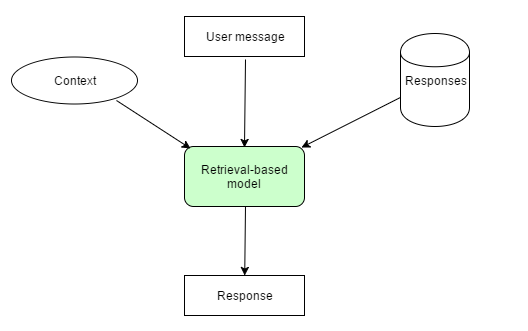
## retrieval-based Model:

Retrieval-based models are more in use at the moment. Several algorithms and APIs are readily available for developers to build chatbots on this architectural model.

This bot considers the message and context of the conversation to deliver the best response from a predefined list of messages.

The aim of the system is to learn how to select the best argument from a pre-defined topic that matches the current user’s response and the history of the conversation.

retrieval-based method is straightforward and guarantee to produce high quality messages, it is limited to only arguments that are available in the dataset and cannot adapt or tailor to every new responses from the users.



All this models are used to find responses based on structured documents. such approaches also depend on existing Q-R pairs as training data. Like other language generation tasks, such as machine translation and paraphrasing, the fluency and naturality of machine generated text is another drawback.

# The way to find responses based on Un structured data :

a novel response retrieval approach**, DocChat**, to find responses based on unstructured documents. For each user utterance, instead of looking for the best Q-R pair or generating a word sequence based on language generation techniques, our method selects a sentence from given documents directly, by ranking all possible sentences based on features designed at different levels of granularity

## An overview of recent chatbots:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| chatbots | IBM watson | Google Assistant | cortana | Alexa | Siri |
|  | * 2006 * Watson uses IBM’s Deep QA software and Approach UIMA (Unstructured information Management Architecture) | * 2016 * It is the most accessible chatbot , as it is available on all Android and IOS devices (Mobile App) and also in chromebooks   (**Chromebooks** are laptops, detachables and tablets powered by Chrome OS: the operating system that is speedy, smart and secure.) (Desktop App)   * Google has it’s own line of google home speakers | * 2014 * Second accessible chatbot * Microsoft's AI, Cortana, comes standard on Windows machines(desktop App), but it's also available for download on Android and iOS(Mobile App). | * 2015 * Third accessible chatbot. * Amazon's smart assistant is accessible through the Amazon Echo line of speakers as well as through Amazon's line of Fire tablets and Fire TV. Of all the AI assistants, Alexa is available on the widest variety of devices in terms of design. | * 2011 * fourth accessible chatbot. * 3-You can access Siri on nearly any Apple device, including its line of laptops(Desktop App), desktops, phones (Mobile App)and tablets, and smartwatches. |

# Conclusion :

According to our customer ,who are developers , we decided to create our chatbot ,which is web-based ,as Web Apps, are accessed via the internet browser and will adapt to whichever device you are viewing them on. They are not native to a particular system, and are not required to be downloaded or installed.

# **Extra:**

If you’re developing a chatbot, you need to choose a bot-building framework. A bot framework is like a workshop full of tools you can use to define your bot’s behavior, give it a personality, teach it what to say, and help it generate valuable responses to user inquiries. It’s where your bot comes into its own. You need to choose the one that makes the most sense for your objectives, and the best way to make an informed choice is to have a solid understanding of different frameworks. In this guide, we cover the nuts and bolts of the most popular bot-building frameworks:

* Amazon Lex
* Botkit
* Botpress
* Dialogflow
* Gupshup Bot Platform
* IBM Watson Assistant
* Microsoft Bot Framework
* Pandorabots
* Rasa
* Wit.ai

|  | **Primary Uses** | **Pricing** | **Languages/SDKs** | **Chatbot Examples** |
| --- | --- | --- | --- | --- |
| **Amazon Lex** | * Device control * Informational * Productivity * Transactional | **Free to start** Monthly fee per requests processed, currently: $0.004 per voice request; $0.00075 per text request | Java, JavaScript, Python, CLI, .NET, Ruby, PHP, Go, CPP, iOS and Android SDKs | * [BuildFax Counts](https://devpost.com/software/buildfax-counts) * [Infor Coleman](https://www.infor.com/products/coleman) * [NASA Rov-E](https://mars.nasa.gov/ask-nasa-mars/#/) |
| **Botkit** | * Commerce * NLP integrations * Productivity | **Free** | Node.js, Botkit SDK | * [Harvard Business Review](https://botlist.co/bots/harvard-business-review) * [HR Slack Bot](https://medium.freecodecamp.org/how-i-built-an-hr-slack-bot-with-node-and-botkit-6b23b81531bb) |
| **Botpress** | * Commerce * Customer service * Financial services | * **Free Tier** * **Pro Tier:** $49.95/month * **Enterprise Tier:** Custom price | Node.js, Botpress SDK | * [Andy Carnegie Bot](https://studio.carnegiemuseums.org/a-chatbot-adventure-c729812571ea) |
| **Dialogflow** | * Commerce * Enterprise * Service | **Free for Standard Edition** Rates of $0.002 to $0.065 per consumption unit in Enterprise Edition | Node.js, Python, Java, Go, Ruby, C#, PHP | * [Domino's](https://dialogflow.com/case-studies/dominos/) * [KLM](https://dialogflow.com/case-studies/klm/) * [Ticketmaster](https://dialogflow.com/case-studies/ticketmaster/) |
| **Gupshup Bot Platform** | * Customer service * Device control * Informational commerce | **Free for 100,000 API calls** $1 for every 1,000 additional API calls Custom pricing for large volumes | Java SDK, Node.js SDK | * [FitCircle](https://www.gupshup.io/developer/resources/downloads/usecases/Embed%20Android%20SDK%20in%20to%20a%20fitness%20app.pdf) * [Tata Sky](https://www.gupshup.io/developer/resources/downloads/usecases/TataSky.pdf) |
| **IBM Watson Assistant** | * Customer service * Enterprise * Commerce | **Five Tiers:** Free tier, $0.0025 per message tier, and three custom pricing tiers | Java, Node.js, Python, Ruby, .NET, Android SDK, OpenWhisk SDK, Salesforce SDK, Unity SDK | * [Autodesk](https://www.ibm.com/case-studies/autodesk-inc) * [Creval Sistemi e Servizi, "Alfredo"](https://www.ibm.com/case-studies/crevalsistemieservizi) * [Woodside](https://www.ibm.com/watson/stories/woodside/) |
| **Microsoft Bot Framework** | * Commerce * Informational * Enterprise | * **Free Tier:** Unlimited messages on standard channels; 10,000 messages/month on premium channels * **S1 Tier:** Unlimited messages on standard channels; $0.50 per 1,000 on premium channels | Node.js, C#, .NET | * [NAVITIME](https://microsoft.github.io/techcasestudies/cognitive%20services/2017/06/28/Navitime.html) * [Super Brand Mall](https://microsoft.github.io/techcasestudies/bot%20framework/2017/06/21/CaaP-SuperBrandMall.html) * [TechnologyOne](https://microsoft.github.io/techcasestudies/bot%20framework/cognitive%20services/2017/06/13/TechnologyOneEducationBot.html) |
| **Pandorabots** | * Customer service * Device control * Entertainment | * **Free tier:** Up to 1,000 messages/month Developer tier: $0.0025 per message * **Developer Tier: $0.0025 per message** * **Enterprise tier:** Custom price | Java, Node.js, Python, Ruby, PHP, Go | * [American Eagle](https://www.facebook.com/business/success/american-eagle-outfitters) * [Coca-Cola vending machine](https://www.coca-colacompany.com/stories/smart-vending-coke-readying-ai-powered-drink-machines) * [Mitsuku](https://www.pandorabots.com/mitsuku/) |
| **Rasa** | * Contextual assistants * Financial * Service * Travel | * **Free Tier** * **Paid Tier** (Rasa Platform) with custom pricing | HTTP API, Python | * [Tia](https://www.asktia.com/app/) * [Yellobot](https://medium.com/rasa-blog/building-with-rasa-yellow-pages-canada-chatbot-7a5d75d052c7) |
| **Wit.ai** | * Commerce * Device control * Informational | **Free** | Node.js, Python, Ruby, HTTP API | * [Car Dost](https://www.facebook.com/RideWidth) * [Meno BOT](https://devpost.com/software/meno-bot) * [Wedding Planner](https://medium.com/@MissAmaraKay/first-chatbot-experience-with-wit-ai-47c2ebddf050) |

# References :

* Paper\_12 Survey\_on\_Chatbot\_Design\_Techniques\_in\_Speech\_Conversation\_Systems.pdf
* Surveyofchatbotsystems.pdf
* Chatbots\_Are\_they\_Really\_Useful.pdf
* [10.1109@ICEBE.2018.00019.pdf](mailto:10.1109@ICEBE.2018.00019.pdf)(A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks)
* 5c49d4a7cba9bf077d97b401ba78aafe693f.pdf(A SURVEY OF VARIOUS CHATBOT IMPLEMENTATION TECHNIQUES)
* anthology/P16-1049(DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents)
* anthology/W18-5215(Dave the debater: a retrieval-based and generative argumentative dialogue agent)