

CHATBOT SONG RECOMMENDATION SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and
Engineering

By

**K MEETUL KUMAR (Reg.no - 39110616)
E SHREE HARSHA (Reg.no - 39110939)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **K MEETUL KUMAR (39110616)** and **E SHREE HARSHA (39110939)** who carried out the Project Phase-2 entitled "**CHATBOT SONG RECOMMENDATION SYSTEM**" under my supervision from January 2023 to April 2023.

Internal Guide

Dr. PREM JACOB, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.



Submitted for Viva-voce Examination held on 20.04.2023

Internal Examiner

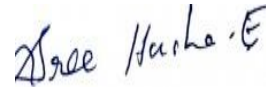
External Examiner

DECLARATION

I, **E Shree Harsha (Reg.no- 39110939)**, hereby declare that the Project Phase-2 Report entitled “**CHATBOT SONG RECOMMENDATION SYSTEM**” done by me under the guidance of **Dr. PREM JACOB, M.E, Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 20-04-23

PLACE: Chennai

A handwritten signature in blue ink that reads "Shree Harsha E".

SIGNATURE OF THE CANDIDATE

iii
ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me with the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. PREM JACOB, M.E., Ph.D.**, for his valuable guidance, suggestions, and constant encouragement that paved the way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways in the completion of the project.

iv
ABSTRACT

The purpose of chatbots is to support and scale business teams in their relations with customers. It could live in any major chat application like Facebook Messenger, Slack, Telegram, Text Messages, etc. Chatbot applications streamline interactions between people and services, enhancing customer experience. At the same time, they offer companies new opportunities to improve the customer engagement process and operational efficiency by reducing the typical cost of customer service. This project is focused on building a custom chatbot that will be your fundamental step of the learning curve of building your own professional chatbots.

But you must be tired of the weird chatbots out there in the world which are made for mainly business purposes. In this project, we would be building an extensive Chatbot service, to which you can talk to. And talking to a chatbot wouldn't be business-driven. It would just be casual conversations. Further, on top of it, the chatbot would also be recommending songs to the user based on the tone of the user. This song recommendation feature employs the use of Last.fm API, very much similar to the popular Spotify API. Also, for tone/emotion analysis of the conversation, we will be using the IBM Tone Analyzer API.

Collaborating with these types of APIs is very much critical as in today's world the popular chatbots do much more than simply have a data-driven conversation; to supplement additional user-oriented features. Also, the reason to choose Python to build the chatbot is that Python boasts a wide array of open-source libraries for chatbots, including scikit-learn and TensorFlow. It is great for small data sets and more simple analyses; also, Python's libraries are much more practical.

Chapter

No TITLE Page No. ABSTRACT 5 1 OBJECTIVE 8

INTRODUCTION TO CHATBOT

2₁ 0 2.1 What is a Chatbot

2.2 Working of Chatbot 11 2.3 Evolution of Chatbot 12 2.4 Common

Uses of Chatbot 13 ³ TYPES OF CHATBOT

3.1.1. Voice Bots 18 3.1.2. Hybrid Chatbot 19 3.1.3. Social Messaging Chatbot 19 3.1.4. Menu-based Chatbot 19

3.1.5. Skills Chatbot 20 3.1.6. Keyboard-based Chatbot 20 3.1.7. Rules-based Chatbot 21 3.1.8. AI-powered Contextual Chatbot 21 3.1.9. Support Chatbot 21 3.1.10. Transactional Chatbot 21 3.2. Limitation of chatbot 22

4 ⁴ Recommender System 17 4.1. Approaches 23

4.1.1. Collaborative filtering 23 4.1.2. Content-based filtering 25 4.1.3. Hybrid recommendations approaches 27

4.2. Technologies

4.2.1. Session-based recommender systems 28 4.2.2. Reinforcement learning for recommender systems 28 4.2.3. Multi-criteria recommender systems 28 4.2.4. Risk-aware recommender systems 29 4.2.5. Mobile recommender systems 29

4.3. Recommender Evaluation 29 4.4. Beyond accuracy 31

4.5. Spotify recommendation	32	4.6. Youtube recommendation	33
5		6.2.1. Future Bots	
Project Literature		References	
		33	
5.1. Architecture			
5.2. Proposed Model	35	37	39
5.2.1. Algorithm Used	39		
5.3. Natural Language Processing	39		
NLP Tasks	40		
5.3.2. NLP Use Cases	41		
5.4. IBM Tone Analyzer	42		
5.5. Anvil Framework	42		
5.6. Last.FM	43		
5.6.1 Features	45		
5.7. Survey	45		
5.8. Tools and Technologies			
6	46		
Conclusion and Future work	47		
6.1. Conclusion	47		
6.2. Future Use	48		

LIST OF FIGURES

Figure No Figure Name Page No.

1.1 Music Chatbot 8 2.1 Introduction to Chatbot 10 2.2 History of Chatbot 11

2.3 Evolution of Chatbot 12 2.4 Chatbot AI 13 3.1 Types of Chatbot 18 4.1

Recommender System 23

4.2 Collaborative filtering 24 4.3 Content-based filtering 25 4.4 Hybrid
recommendations approaches 27 4.5 Recommender Evaluation 30 4.6 *Spotify
recommendation* 32 4.7 *Youtube recommendation* 33 5.1 Chatbot adoption
across different industries 34 5.2 Architecture Diagram 35 5.3 Potential Benefits
of Chatbots 36 5.4 Flowchart for Song Evaluation 38 5.5 Mood Classification
44 6.1 Chatbot Technology adoption 46

CHAPTER 1: Objective



Fig 1.1

In this project, we would be combining multiple services and open-source tools to make a Chatbot that recommends songs based on the resonance of the conversation which the user is having with the chatbot.

A chatbot is a software application used to conduct an online chat conversation via text or text-to-speech, it's providing direct contact with a live human agent.

We can also say that a chatbot is an intelligent piece of software that is capable of communicating and performing actions similar to a human.

It provides an interactive platform to communicate with users.

Chatbots have varying levels of complexity, being either stateless or stateful. Stateless chatbots approach each conversation as if interacting with a new user.

In contrast, stateful chatbots can review past interactions and frame new responses in context. • Adding a chatbot to a service or sales department requires low or no coding. • Many chatbot service providers allow developers to build conversational user interfaces for third-party business applications. • In this project we created a simple chatbot that will be used to answer frequently asked questions.

discuss more the critical components of chatbot architecture:

Question and Answer System: As the name suggests, the Q&A system is responsible for answering customers' frequently asked questions. The question is interpreted by the Q&A system, which then replies with appropriate responses from the knowledge base.

It consists of the following elements:

- Manual Training:** Manual training entails the domain specialist compiling a list of commonly asked user questions and mapping out the answers. It enables the chatbot to identify the most relevant questions' answers rapidly.
- Automated Training:** Automated training entails sending business documents to the chatbot, such as policy documents and other Q&A-type documents, and instructing it to train itself. From these documents, the engine generates a list of questions and responses. The chatbot would then be able to respond with confidence.

2) **Environment:** - The environment is mainly responsible for contextualizing users' messages using natural language processing (NLP). The NLP Engine is the central component of the chatbot architecture. It interprets what users are saying at any given time and turns it into organized inputs that the system can process. The NLP engine uses advanced machine learning algorithms to determine the user's intent and then match it to the chatbot's supported intents list. NLP Engine has two components:

- **Intent Classifier:** An intent classifier maps between what a user asks and the type of action performed by the software.
- **Entity Extractor:** The entity extractor is responsible for identifying keywords from the user's query that help determine what the user is looking for. An NLP engine can also be extended to include feedback mechanisms and policy learning for better overall learning of the NLP engine.

□

Feedback Mechanism: This includes the feedback for the chatbot provided by the users. This part of learning can be incorporated into the chatbot itself. Here, the user rates the interaction at the end of the conversation. It encourages the bot to learn from its mistakes and improve in future interactions.

Policy Learning: Policy learning is a broad framework wherein the bot is trained to create a network of happy paths in the conversation flow that increases overall end-user satisfaction.

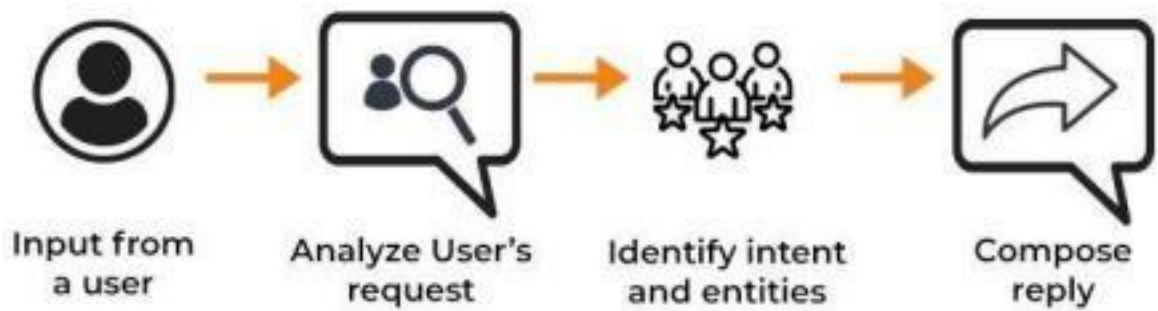


Fig 2.1

2.1. What is a Chatbot?

A chatbot is an artificial intelligence (AI) program that simulates interactive human conversation by using key pre-calculated user phrases and auditory or text-based signals. Chatbots are frequently used by organizations to provide 24-hour customer relationship management (CRM) services. This type of software bot can also be used as an intelligent virtual assistant.

A chatbot is also known as an artificial conversational entity (ACE), chat robot, talk bot, chatterbot, or chatterbox.

2.2. Working of Chatbots



Fig 2.2

Driven by AI, automated rules, natural language processing (NLP), and machine learning (ML), chatbots process data to deliver responses to requests of all kinds. There are two main types of chatbots. Task-oriented (declarative) chatbots are single-purpose programs that focus on performing one function. Using rules, NLP, and very little ML, they generate automated but conversational responses to user inquiries. Interactions with these chatbots are highly specific and structured and are most applicable to support and service functions—think robust, interactive FAQs. Task-oriented chatbots can handle common questions, such as queries about hours of business or simple transactions that don't involve a variety of variables. Though they do use NLP so end users can experience them in a conversational way, their capabilities are fairly basic. These are currently the most commonly used chatbots.

Data-driven and predictive (conversational) chatbots are often referred to as virtual assistants or digital assistants, and they are much more sophisticated, interactive, and personalized than task-oriented chatbots. These chatbots are contextually aware and leverage natural-language understanding (NLU), NLP, and ML to learn as they go. They apply predictive intelligence and analytics to enable personalization based on user profiles and past user behavior. Digital assistants can learn a user's preferences over time, provide recommendations, and even anticipate needs. In addition to monitoring data and intent, they can initiate conversations. Apple's Siri and Amazon's Alexa are examples of consumer-oriented, data-driven, predictive chatbots.

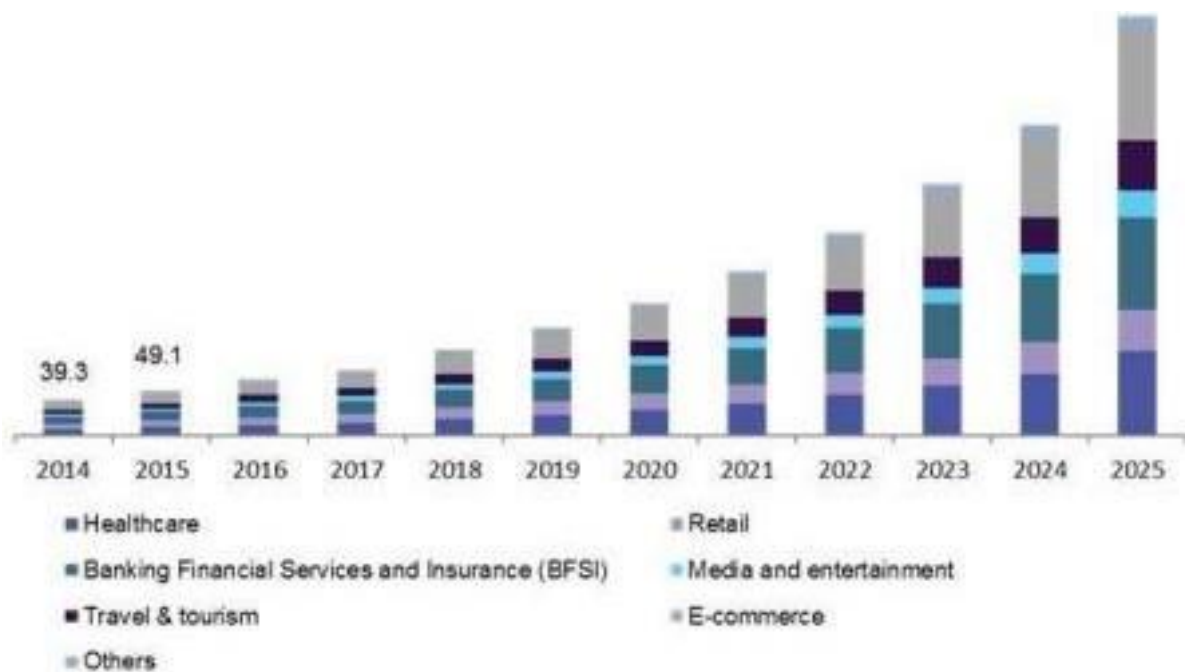


Fig 2.3

The origin of the chatbot arguably lies in Alan Turing's 1950s vision of intelligent machines. Artificial intelligence, the foundation for chatbots, has progressed since that time to include super intelligent supercomputers such as IBM Watson. The original chatbot was the phone tree, which led phone-in customers on an often cumbersome and frustrating path of selecting one option after another to wind their way through an automated customer service model.

Enhancements in technology and the growing sophistication of AI, ML, and NLP evolved this model into pop-up, live, onscreen chats. And the evolutionary journey has continued. With today's digital assistants, businesses can scale AI to provide much more convenient and effective interactions between companies and customers—directly from customers' digital devices.



Fig 2.4

Chatbots are frequently used to improve the IT service management experience, which delves towards self-service and automating processes offered to internal staff. With an intelligent chatbot, common tasks such as password updates, system status, outage alerts, and knowledge management can be readily automated and made available 24/7, while broadening access to commonly used voice and text-based conversational interfaces.

Messaging apps

Many company's chatbots run on messaging apps or simply via SMS. They are used for B2C customer service, sales, and marketing.

In 2016, Facebook Messenger allowed developers to place chatbots on their platforms. There were 30,000 bots created for Messenger in the first six months, rising to 100,000 by September 2017.

Since September 2017, this has also been part of a pilot program on WhatsApp. Airlines KLM and Aeroméxico both announced their participation in the testing both airlines had previously launched customer services on the Facebook Messenger platform.

The bots usually appear as one of the user's contacts, but can sometimes act as participants in a group chat.

Many banks, insurers, media companies, e-commerce companies, airlines, hotel chains, retailers, healthcare providers, government entities, and restaurant chains have used chatbots to answer simple questions and increase customer engagement. for promotion, and to offer additional ways to order from them.

As part of company apps and websites

Previous generations of chatbots were present on company websites, e.g., Ask Jenn from Alaska Airlines which debuted in 2008 or Expedia's virtual customer service agent which launched in 2011. The newer generation of chatbots includes IBM Watson-powered "Rocky", introduced in February 2017 by the New York City-based e-commerce company Rare Carat to provide information to prospective diamond buyers.

Chatbot sequences

Used by marketers to script sequences of messages, very similar to an autoresponder sequence. Such sequences can be triggered by user opt-in or the use of keywords within user interactions. After a trigger occurs a sequence of messages is delivered until the next anticipated user response. Each user response is used in the decision tree to help the chatbot navigate the response sequences to deliver the correct response message.

Company internal platforms

Other companies explore ways they can use chatbots internally, for example for Customer Support, Human Resources, or even in Internet-of-Things (IoT) projects. Overstock.com, for one, has reportedly launched a chatbot named Mila to automate certain simple yet time-consuming processes when requesting sick leave. Other large companies such as Lloyds Banking Group, Royal Bank of Scotland, Renault, and Citroën are now using automated online assistants instead of call centers with humans to provide a first point of contact. A SaaS chatbot business ecosystem has been steadily growing since the F8 Conference when Facebook's Mark Zuckerberg unveiled that Messenger would allow chatbots into the app. In large companies, like in hospitals and aviation organizations, IT architects are designing reference architectures for Intelligent Chatbots that are used to unlock and share knowledge and experience in the organization more efficiently and reduce the errors in answers from expert service desks significantly. These Intelligent Chatbots make use of all kinds of artificial intelligence like image moderation and natural-language understanding (NLU), natural-language generation (NLG), machine learning, and deep learning.

Customer service

Many high-tech banking organizations are looking to integrate automated AI-based solutions such as chatbots into their customer service in order to provide faster and cheaper assistance to their clients who are becoming increasingly comfortable with technology. In particular, chatbots can efficiently conduct a dialogue, usually replacing other communication tools such as email, phone, or SMS. In banking, their major application is related to quick customer service answering common requests,

as well as transactional support.

Several studies report significant reduction in the cost of customer services, expected to lead to billions of dollars of economic savings in the next ten years. In 2019, Gartner predicted that by 2021, 15% of all customer service interactions globally will be handled completely by AI. A study by Juniper Research in 2019

estimates retail sales resulting from chatbot-based interactions will reach \$112 billion by 2023.

Since 2016, when Facebook allowed businesses to deliver automated customer support, e-commerce guidance, content, and interactive experiences through chatbots, a large variety of chatbots were developed for the Facebook Messenger platform.

In 2016, Russia-based Tochka Bank launched the world's first Facebook bot for a range of financial services, including the possibility of making payments.

In July 2016, Barclays Africa also launched a Facebook chatbot, making it the first bank to do so in Africa.

France's third-largest bank by total assets Société Générale launched its chatbot called SoBot in March 2018. While 80% of users of the SoBot expressed their satisfaction after having tested it, Société Générale deputy director Bertrand Cozzarolo stated that it will never replace the expertise provided by a human advisor.

The advantages of using chatbots for customer interactions in banking include cost reduction, financial advice, and 24/7 support.

Healthcare

Chatbots are also appearing in the healthcare industry. A study suggested that physicians in the United States believed that chatbots would be most beneficial for scheduling doctor appointments, locating health clinics, or providing medication information.

Whatsapp has teamed up with the World Health Organisation (WHO) to make a chatbot service that answers users' questions on COVID-19.

In 2020, The Indian Government launched a chatbot called MyGov Corona Helpdesk, that worked through Whatsapp and helped people access information about the Coronavirus (COVID-19) pandemic.

Certain patient groups are still reluctant to use chatbots. A mixed-methods study showed that people are still hesitant to use chatbots for their healthcare due to a poor understanding of the technological complexity, the lack of empathy, and concerns about cyber-security. The analysis showed that while 6% had heard of a health chatbot and 3% had the experience using it, 67% perceived themselves as likely to use one within 12 months. The majority of participants would use a health chatbot for seeking general health information (78%), booking a medical

appointment (78%), and looking for local health services (80%). However, a health chatbot was perceived as less suitable for seeking results of medical tests and seeking specialist advice such as sexual health. The analysis of attitudinal variables showed that most participants reported their preference for discussing their health with doctors (73%) and having access to reliable and accurate health information (93%). While 80% were curious about new technologies that could improve their health, 66% reported only seeking a doctor when experiencing a health problem and 65% thought that a chatbot was a good idea. Interestingly, 30% reported dislike

about talking to computers, 41% felt it would be strange to discuss health matters with a chatbot and about half were unsure if they could trust the advice given by a chatbot. Therefore, perceived trustworthiness, individual attitudes towards bots, and dislike for talking to computers are the main barriers to health chatbots.

Politics

In New Zealand, the chatbot SAM – short for Semantic Analysis Machine (made by Nick Gerritsen of Touchtech) – has been developed. It is designed to share its political thoughts, for example on topics such as climate change, healthcare and education, etc. It talks to people through Facebook Messenger.

In 2022, the chatbot "Leader Lars" or "Leder Lars" was nominated for The Synthetic Party to run in the Danish parliamentary election, and was built by the artist collective Computer Lars. Leader Lars differed from earlier virtual politicians by leading a political party and by not pretending to be an objective candidate. This chatbot engaged in critical discussions on politics with users from around the world.

In India, the state government has launched a chatbot for its Aaple Sarkar platform, which provides conversational access to information regarding public services managed.

Toys

Chatbots have also been incorporated into devices not primarily meant for computing, such as toys.

Hello Barbie is an Internet-connected version of the doll that uses a chatbot provided by the company ToyTalk, which previously used the chatbot for a range of smartphone-based characters for children. These characters' behaviors are constrained by a set of rules that in effect emulate a particular character and produce a storyline.

The My Friend Cayla doll was marketed as a line of 18-inch (46 cm) dolls which uses speech recognition technology in conjunction with an Android or iOS mobile app to recognize the child's speech and have a conversation. It, like the Hello Barbie doll, attracted controversy due to vulnerabilities with the doll's Bluetooth stack and its use of data collected from the child's speech.

IBM's Watson computer has been used as the basis for chatbot-based educational toys for companies such as CogniToys intended to interact with children for educational purposes.

Malicious use

Malicious chatbots are frequently used to fill chat rooms with spam and advertisements, by mimicking human behavior and conversations or to entice people into revealing personal information, such as bank account numbers. They were commonly found on Yahoo! Messenger, Windows Live Messenger, AOL Instant Messenger, and other instant messaging protocols. There has also been a published report of a chatbot used in a fake personal ad on a dating service's website.

Tay, an AI chatbot that learns from previous interactions, caused major controversy due to it being targeted by internet trolls on Twitter. The bot was exploited, and after 16 hours began to send extremely offensive Tweets to users. This suggests that although the bot learned effectively from experience, adequate protection was not put in place to prevent misuse.

If a text-sending algorithm can pass itself off as a human instead of a chatbot, its message would be more credible. Therefore, human-seeming chatbots with well crafted online identities could start scattering fake news that seems plausible, for instance making false claims during an election. With enough chatbots, it might be even possible to achieve artificial social proof.

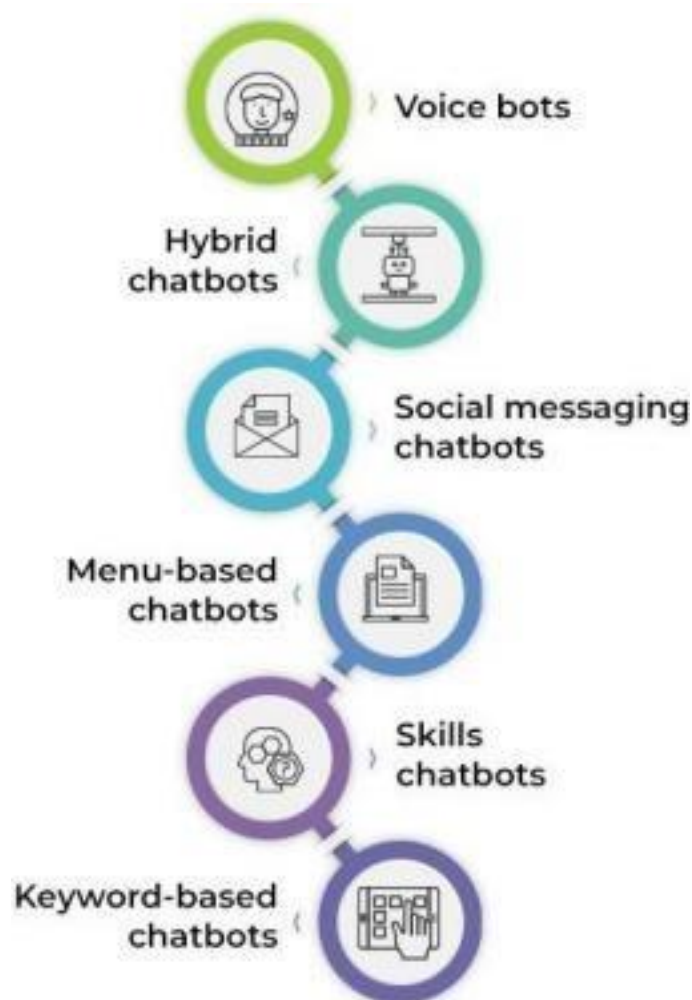


Fig 3.1

Hundreds of thousands of businesses worldwide are developing diverse forms of chatbots intending to enhance customer service. This section explains the various types of chatbots, what they are used for, and which chatbot software could be the most beneficial to your company.

3.1.1. Voice bots

A voice bot is a voice-to-text and text-to-speech communication channel powered by AI and natural language understanding (NLU). AI technology aids in identifying key speech signals and determining the optimal conversational response. The text-to-speech (TTS) engine subsequently completes the interaction by converting the message into audio or voice.

These bots are programmed to complete the entire speech comprehension and

response process in a human-like manner. Voice assistants or voice chatbots provide a sophisticated model of communication that one may quickly implement into various customer service tools, including interactive voice response (IVR),

self-service, and online information basis.

3.1.2. Hybrid Chatbots

A hybrid chatbot is a harmonious blend of chatbot and live chat that combines the best of both worlds. A customer service representative will be available in live chat to answer any customer's questions, which may be too complex or nuanced for automation alone.

An AI component in a chatbot replicates conversations based on how it is programmed and the needs of the conversation. On the other hand, a hybrid chatbot will initiate an automated chat conversation and attempt to resolve the user's query as quickly and simply as possible. If it does not function as expected, a customer service representative can intervene at any moment or in the subject matter area where the chatbot cannot complete the task.

3.1.3. Social messaging Chatbots

With the rise of new social media interfaces, organizations can now deploy an AI algorithm across all of their customer's preferred messaging platforms. This includes Facebook Messenger, Twitter, and Instagram, as well as messaging apps like WhatsApp and WeChat. It enables a more pleasant online experience for customers and increased engagement for the company – all without adding to a contact center's workload.

3.1.4. Menu-based chatbots

The most rudimentary type of chatbot in use is one that is based on menu-driven navigation. Most of the time, these chatbots follow a fixed tree that is displayed to the consumer in the form of clickable buttons. These chatbots (like the automated dial pad menus on telephones that we use regularly) ask the user to make several choices and click on suitable options to get to the final solution.

Although these chatbots are adequate for addressing frequently asked questions, which account for most support requests, they may fall short in more complicated scenarios. If there are too many elements or too much expertise at play, the menu-based chatbot may not be able to help users arrive at the correct response. It is also

important to note that menu-based chatbots are the slowest to deliver genuine value to the consumer, but they are simple and affordable to get started.

3.1.5. Skills chatbots

A skills chatbot is another kind of bot that can perform a specific set of tasks, once you have extended its capabilities using pre-defined skills software. For example, the chatbot may be able to provide weather information, turn off your room lights when connected to a smart home appliance, order groceries online, etc. With access to the skill's source code, developers can construct their own skills chatbots and integrate them with other platforms.

3.1.6. Keyword-based chatbots

Keyword-based chatbots can listen to what visitors enter and answer correctly, unlike menu-based chatbots. These chatbots use customizable keywords and NLP to detect action triggers in the conversation to understand how to respond appropriately to the consumer. However, when faced with many similar inquiries, these chatbots may fall short. The chatbots could begin to falter if there are keyword repetitions across numerous associated inquiries.

That is why chatbots that combine keyword identification and menu or button based navigation are becoming increasingly popular. If the keyword detection functionality fails or the user needs additional help finding an answer, such chatbots give users the option of directly entering commands via clickable navigation buttons. This is an effective workaround when the bot cannot detect keywords in the typed input

3.1.7. Rules-based chatbots

A rule-based chatbot is ideal for companies that already know the types of inquiries their customers will ask. Chat flows are created by using if/then logic, and you must first establish the chatbot's language requirements. Conditions for

evaluating words, word structure, synonyms, and more are the essential tenets of its functionality. Customers will receive prompt assistance if an inbound inquiry falls within these parameters.

3.1.8. *AI-powered contextual chatbots*

Contextual chatbots can grasp the context of a chat and determine the correct meaning of the user inquiry. It can also recall prior interactions and use that information to maintain relevance while interacting with repeat customers. Contextual bots can guarantee that repeat users have a consistent experience. Furthermore, it may retain information on user intent gathered across numerous platforms and channels, ensuring that the conversation's context matches the needs of the consumer at every touchpoint.

3.1.9. *Support chatbots*

Support chatbots are conversational systems designed solely to provide customer support and post-purchase services. Unlike bots on social media or websites, they do not share offers, promos, or other customer engagement materials. This type of chatbot is typically found on self-service portals and online documentation, where users might come to receive support and help. Support chatbots are widely used for internal purposes, including answering HR queries, raising IT tickets, submitting employee documents, etc.

3.1.10. *Transactional bots*

Transactional chatbots can help organizations strengthen their sales and marketing efforts, whether for appointment scheduling, lead generation, or payment collection. Users can conduct transactions directly while conversing with the chatbot without human intervention.

21

Its biggest benefit is the ability to enable transactions and drive business 24/7/365. As a result, a transactional chatbot differs from other types of bots, like informational or support bots. Its focus is to complete a transaction and streamline the user experience by offering a quick and easy channel for a single purpose. It is designed to handle a small number of specialized tasks.

3.2. *Limitations of chatbots*

The creation and implementation of chatbots is still a developing area, heavily related to , so the provided solutions, while possessing obvious advantages, have some important limitations in terms of functionalities and use cases. However, this is changing over time.

The most common limitations are listed below:

As the input/output database is fixed and limited, chatbots can fail while dealing artificial intelligence and machine learning with an unsaved query.

- A chatbot's efficiency highly depends on language processing and is limited because of irregularities, such as accents and mistakes.
- Chatbots are unable to deal with multiple questions at the same time and so conversation opportunities are limited.
- Chatbots require a large amount of conversational data to train. Generative models, which are based on deep learning algorithms to generate new responses word by word based on user input, are usually trained on a large dataset of natural-language phrases.
- Chatbots have difficulty managing non-linear conversations that must go back and forth on a topic with a user.
- As it happens usually with technology-led changes in existing services, some consumers, more often than not from older generations, are uncomfortable with chatbots due to their limited understanding, making it obvious that their requests are being dealt with by machines.

3.3. Chatbots and jobs

Chatbots are increasingly present in businesses and often are used to automate tasks that do not require skill-based talents. With customer service taking place via messaging apps as well as phone calls, there are growing numbers of use-cases where chatbot deployment gives organizations a clear return on investment. Call center workers may be particularly at risk from AI-driven chatbots.

3.4. Chatbot jobs

Chatbot developers create, debug, and maintain applications that automate customer services or other communication processes. Their duties include reviewing and simplifying code when needed. They may also help companies implement bots in their operations.

Chapter 4: Recommender system

A **recommender system**, or a **recommendation system** (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that provide suggestions for items that are most pertinent to a particular user. Typically, the suggestions refer to various decision-making processes, such as what product to purchase, what music to listen to, or what online news to read. Recommender systems are particularly useful when an individual

needs to choose an item from a potentially overwhelming number of items that a service may offer.

Recommender systems are used in a variety of areas, with commonly recognized examples taking the form of playlist generators for video and music services, product recommenders for online stores, content recommenders for social media platforms, and open web content recommenders. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating. Recommender systems have also been developed to explore research articles and experts, collaborators, and financial services

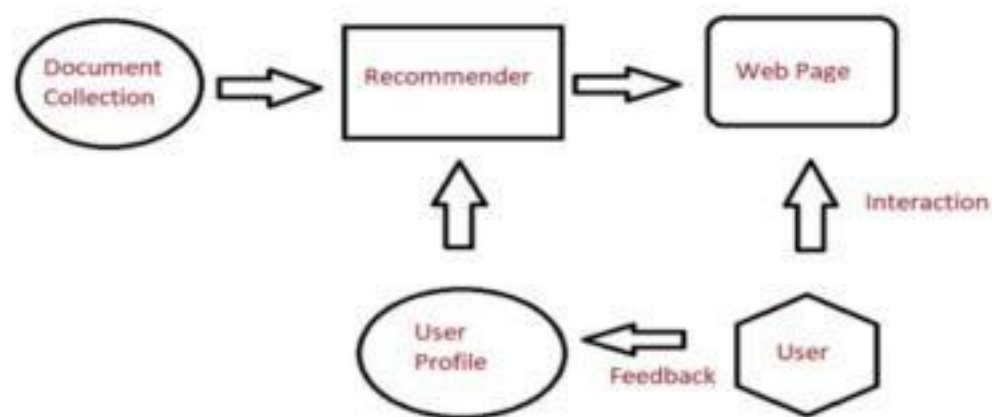


Fig 4.1

4.1. Approaches

4.1.1. Collaborative filtering

One approach to the design of recommender systems that has wide use is collaborative filtering. Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The system generates recommendations using only information about rating profiles for different users or items. By locating peer users/items with a rating history similar to the current user or item, they generate recommendations using this neighborhood. Collaborative filtering methods

are classified as memory-based and model-based. A well-known example of memory-based approaches is the user-based algorithm, while that of model-based approaches is Matrix factorization (recommender systems).

A key advantage of the collaborative filtering approach is that it does not rely on machine-analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. For example, the k-nearest neighbor (k-NN) approach and the Pearson Correlation as first implemented by Allen.

When building a model from a user's behavior, a distinction is often made between

explicit and implicit forms of data collection.

Examples of explicit data collection include the following:

- Asking a user to rate an item on a sliding scale.
- Asking a user to search.
- Asking a user to rank a collection of items from favorite to least favorite.
- Presenting two items to a user and asking him/her to choose the better one of them.
- Asking a user to create a list of items that he/she likes (see Rocchio classification or other similar techniques).

Examples of implicit data collection include the following:

- Observing the items that a user views in an online store.
- Analyzing item/user viewing times.
- Keeping a record of the items that a user purchases online.
- Obtaining a list of items that a user has listened to or watched on his/her computer.
- Analyzing the user's social network and discovering similar likes and dislikes.



Fig 4.2

Collaborative filtering approaches often suffer from three problems: cold start, scalability, and sparsity.

- **Cold start:** For a new user or item, there isn't enough data to make accurate recommendations. Note: one commonly implemented solution to this problem is the Multi-armed bandit algorithm.
- **Scalability:** There are millions of users and products in many of the environments

in which these systems make recommendations. Thus, a large amount of computation power is often necessary to calculate recommendations.

- **Sparsity:** The number of items sold on major e-commerce sites is extremely large. The most active users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings.

One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's recommender system.

Many social networks originally used collaborative filtering to recommend new friends, groups, and other social connections by examining the network of connections between a user and their friends. Collaborative filtering is still used as part of hybrid systems.

4.1.2. Content-based filtering

Another common approach when designing recommender systems is **content based filtering**. Content-based filtering methods are based on a description of the item and a profile of the user's preferences. These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Content-based recommenders treat recommendation as a user specific classification problem and learn a classifier for the user's likes and dislikes based on an item's features.

In this system, keywords are used to describe the items, and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items similar to those that a user liked in the past or is examining in the present. It does not rely on a user sign-in mechanism to generate this often temporary profile. In particular, various candidate items are compared with items previously rated by the user, and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research.

To create a user profile, the system mostly focuses on two types of information:

1. A model of the user's preference.

2. A history of the user's interaction with the recommender system.

Basically, these methods use an item profile (i.e., a set of discrete attributes and features) characterizing the item within the system. To abstract the features of the items in the system, an item presentation algorithm is applied. A widely used algorithm is the tf-idf representation (also called vector space representation). The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian

Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item.

CONTENT-BASED FILTERING

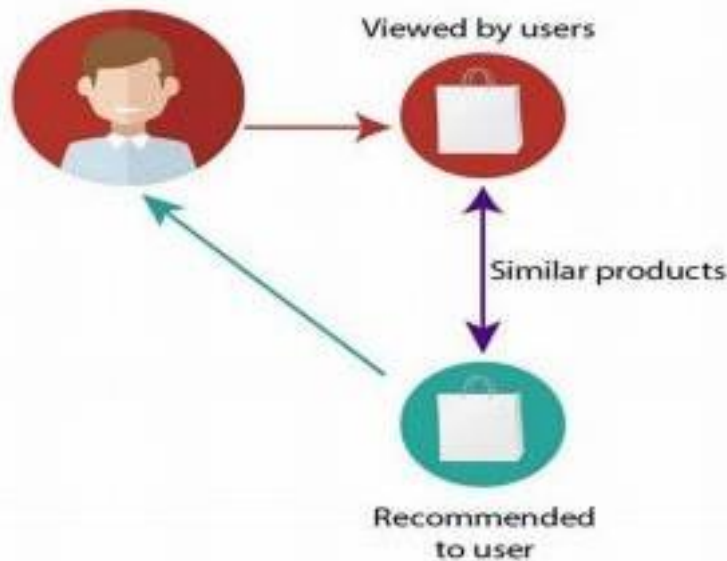


Fig 4.3

A key issue with content-based filtering is whether the system can learn user preferences from users' actions regarding one content source and use them across other content types. When the system is limited to recommending content of the same type as the user is already using, the value from the recommendation system is significantly less than when other content types from other services can be recommended. For example, recommending news articles based on news browsing is useful. Still, it would be much more useful when music, videos, products, discussions, etc., from different services, can be recommended based on news browsing. To overcome this, most content-based recommender systems now use some form of hybrid system.

Content-based recommender systems can also include opinion-based recommender systems. In some cases, users are allowed to leave text reviews or feedback on the items. These user-generated texts are implicit data for the recommender system because they are potentially rich resources of both features/aspects of the item and users' evaluation/sentiment of the item. Features extracted from the user-generated reviews are improved meta-data of items, because as they also reflect aspects of the item like meta-data, extracted features are widely concerned by the users. Sentiments extracted from the reviews can be seen as users' rating scores on the corresponding features. Popular approaches of opinion-based recommender systems utilize various techniques including text mining, information retrieval, sentiment analysis (see also Multimodal sentiment analysis), and deep learning.

4.1.3. Hybrid recommendations approaches

Most recommender systems now use a hybrid approach, combining collaborative filtering, content-based filtering, and other approaches. There is no reason why several different techniques of the same type could not be hybridized. Hybrid approaches can be implemented in several ways: by making

content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model. Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrated that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem, as well as the knowledge engineering bottleneck in knowledge-based approaches.

Netflix is a good example of the use of hybrid recommender systems. The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering).

Fig 4.4

Some hybridization techniques include:

- **Weighted:** Combining the score of different recommendation components numerically.
- **Switching:** Choosing among recommendation components and applying the selected one.
- **Mixed:** Recommendations from different recommenders are presented together to give the recommendation.
- **Feature Combination:** Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.
- **Feature Augmentation:** Computing a feature or set of features, which is then part of the input to the next technique.

- **Cascade:** Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.
- **Meta-level:** One recommendation technique is applied and produces some sort

of model, which is then the input used by the next technique

4.2. Technologies

4.2.1. Session-based recommender systems

These recommender systems use the interactions of a user within a session to generate recommendations. Session-based recommender systems are used on Youtube and Amazon. These are particularly useful when the history (such as past clicks, and purchases) of a user is not available or not relevant in the current user session. Domains, where session-based recommendations are particularly relevant, include video, e-commerce, travel, music and more. Most instances of session based recommender systems rely on the sequence of recent interactions within a session without requiring any additional details (historical, demographic) of the user. Techniques for session-based recommendations are mainly based on generative sequential models such as Recurrent Neural Networks, Transformers, and other deep learning-based approaches

4.2.2. Reinforcement learning for recommender systems

The recommendation problem can be seen as a special instance of a reinforcement learning problem whereby the user is the environment upon which the agent, the recommendation system acts upon in order to receive a reward, for instance, a click or engagement by the user. One aspect of reinforcement learning that is of particular use in the area of recommender systems is the fact that the models or policies can be learned by providing a reward to the recommendation agent. This is in contrast to traditional learning techniques which rely on supervised learning approaches that are less flexible, reinforcement learning recommendation techniques allow to potentially train models that can be optimized directly on metrics of engagement, and user interest.

4.2.3. Multi-criteria recommender systems

Multi-criteria recommender systems (MCRS) can be defined as recommender systems that incorporate preference information upon multiple criteria. Instead of developing recommendation techniques based on a single criterion value, the overall preference of user u for item I , these systems try to predict a rating for unexplored items of u by exploiting preference information on multiple criteria that affect this overall preference value. Several researchers approach MCRS as a multi criteria decision-making (MCDM) problem and apply MCDM methods and techniques to implement MCRS systems. See this chapter for an extended introduction.

4.2.4. Risk-aware recommender systems

The majority of existing approaches to recommender systems focus on recommending the most relevant content to users using contextual information, yet do not take into account the risk of disturbing the user with unwanted notifications. It

is important to consider the risk of upsetting the user by pushing recommendations in certain circumstances, for instance, during a professional meeting, early morning, or late at night. Therefore, the performance of the recommender system depends in part on the degree to which it has incorporated the risk into the recommendation process. One option to manage this issue is *DRARS*, a system that models the context-aware recommendation as a bandit problem. This system combines a content-based technique and a contextual bandit algorithm.

4.2.5. Mobile recommender systems

Mobile recommender systems make use of internet-accessing smartphones to offer personalized, context-sensitive recommendations. This is a particularly difficult area of research as mobile data is more complex than data that recommender systems often have to deal with. It is heterogeneous, noisy, requires spatial and temporal auto-correlation, and has validation and generality problems.

There are three factors that could affect the mobile recommender systems and the accuracy of prediction results: the context, the recommended method, and privacy. Additionally, mobile recommender systems suffer from a transplantation problem – recommendations may not apply in all regions (for instance, it would be unwise to recommend a recipe in an area where all of the ingredients may not be available).

One example of a mobile recommender system is the approaches taken by companies such as Uber and Lyft to generate driving routes for taxi drivers in a city. This system uses GPS data of the routes that taxi drivers take while working, which includes location (latitude and longitude), time stamps, and operational status (with or without passengers). It uses this data to recommend a list of pickup points along a route, with the goal of optimizing occupancy times and profits.

4.3. Recommender Evaluation

Evaluation is important in assessing the effectiveness of recommendation algorithms. To measure the [effectiveness](#) of recommender systems, and compare different approaches, three types of [evaluations](#) are available: user studies, [online evaluations \(A/B tests\)](#), and offline evaluations.

The commonly used metrics are the [mean squared error](#) and [root mean squared error](#), the latter having been used in the Netflix Prize. The information retrievals metrics such as [precision and recall](#) or [DCG](#) are useful to assess the quality of a recommendation method. Diversity, novelty, and coverage are also considered important aspects of evaluation. However, many of the classic evaluation measures are highly criticized.

Fig 4.5

Evaluating the performance of a recommendation algorithm on a fixed test dataset will always be extremely challenging as it is impossible to accurately predict the reactions of real users to the recommendations. Hence any metric that computes the effectiveness of an algorithm in offline data will be imprecise.

User studies are rather a small scale. A few dozens or hundreds of users are presented recommendations created by different recommendation approaches, and then the users judge which recommendations are best.

In A/B tests, recommendations are shown to typically thousands of users of a real product, and the recommender system randomly picks at least two different recommendation approaches to generate recommendations. The effectiveness is measured with implicit measures of effectiveness such as [conversion rate](#) or [click through rate](#).

Offline evaluations are based on historic data, e.g. a dataset that contains information about how users previously rated movies.

The effectiveness of recommendation approaches is then measured based on how well a recommendation approach can predict the users' ratings in the dataset. While a rating is an explicit expression of whether a user liked a movie, such information is not available in all domains. For instance, in the domain of citation recommender systems, users typically do not rate a citation or recommended article. In such cases, offline evaluations may use implicit measures of effectiveness. For instance, it may be assumed that a recommender system is effective that is able to recommend as many articles as possible that are contained in a research article's reference list. However, this kind of offline evaluations is seen critical by many researchers. For instance, it has been shown that results of offline evaluations have low correlation with results from user studies or A/B tests. A dataset popular for offline evaluation has been shown to contain duplicate data and thus to lead to wrong conclusions in the evaluation of algorithms. Often, results of so-called offline evaluations do not correlate with actually assessed user-satisfaction. This is probably because offline

highly influenced by the outputs of the online recommendation module. Researchers have concluded that the results of offline evaluations should be viewed critically.

4.4. Beyond accuracy

Typically, research on recommender systems is concerned with finding the most accurate recommendation algorithms. However, there are a number of factors that are also important.

- **Diversity** – Users tend to be more satisfied with recommendations when there is a higher intra-list diversity, e.g. items from different artists.
- **Recommender persistence** – In some situations, it is more effective to re-show recommendations, or let users re-rate items, than showing new items. There are several reasons for this. Users may ignore items when they are shown for the first time, for instance, because they had no time to inspect the recommendations carefully.
- **Privacy** – Recommender systems usually have to deal with privacy concerns because users have to reveal sensitive information. Building [user profiles](#) using collaborative filtering can be problematic from a privacy point of view. Many European countries have a strong culture of [data privacy](#), and every attempt to introduce any level of user [profiling](#) can result in a negative customer response. Much research has been conducted on ongoing privacy issues in this space. The [Netflix Prize](#) is particularly notable for the detailed personal information released in its dataset. Ramakrishnan et al. have conducted an extensive overview of the trade-offs between personalization and privacy and found that the combination of weak ties (an unexpected connection that provides serendipitous recommendations) and other data sources can be used to uncover identities of users in an anonymized dataset.
- **User demographics** – Beel et al. found that user demographics may influence how satisfied users are with recommendations. In their paper they show that elderly users tend to be more interested in recommendations than younger users.
- **Robustness** – When users can participate in the recommender system, the issue of fraud must be addressed.
- **Serendipity** – [Serendipity](#) is a measure of "how surprising the recommendations are". For instance, a recommender system that recommends milk to a customer in a grocery store might be perfectly accurate, but it is not a good recommendation because it is an obvious item for the customer to buy. "[Serendipity] serves two purposes: First, the chance that users lose interest because the choice set is too uniform decreases. Second, these items are needed for algorithms to learn and improve themselves".
- **Trust** – A recommender system is of little value for a user if the user does not trust the system. Trust can be built by a recommender system by explaining how it generates recommendations, and why it recommends an item.

- **Labelling** – User satisfaction with recommendations may be influenced by the labeling of the recommendations. For instance, in the cited study [click-through rate](#) (CTR) for recommendations labeled as "Sponsored" were lower (CTR=5.93%) than CTR for identical recommendations labeled as "Organic" (CTR=8.86%). Recommendations with no label performed best (CTR=9.87%) in that study.

4.5. Spotify recommendation

The Spotify recommender system is an extremely complex and intricate system, with dozens (if not hundreds) of algorithms and ML models employed across various levels, all working together to create one of the most advanced recommendation experiences on the music streaming market. This system has been developed and iterated on for close to 12 years now — growing in size, capabilities, and complexity. Yet, as you can probably see, it is far from unexplainable. Even without having direct documentation describing the composition of the recommendation engine and all the secret ingredients, we can get a pretty good understanding of its main parts and the governing principles behind them.

From the music industry perspective, it is possible to use that knowledge to optimize the artist's profile within that recommender system. A meaningful, well-educated algorithmic strategy can maximize your chances of making it onto algorithmic playlists and help ensure that the engine serves your music to the right audiences, amplifying discovery and turning casual listeners into fans.

The satisfaction prediction produced by the model is then, in turn, used as the reward for the algorithm that would compose Discover Weekly playlists, thus building a comprehensive reward system that doesn't rely on siloed, potentially ambiguous user signals.

Fig 4.6

4.6. Youtube recommendation

The YouTube algorithm selects videos for viewers with two goals in mind: finding the right video for each viewer, and enticing them to keep watching. When we talk about

“the algorithm,” we’re talking about three related but slightly different selection or discovery systems:

- one that selects videos for the YouTube homepage;
- one that ranks results for any given search; and
- one that selects suggested videos for viewers to watch next.

Youtube says that homepage and suggested videos are usually the top sources of traffic for most channels. Except for explainer or instructional videos (i.e., “how to tune up a bicycle”), which often see the most traffic from search, instead.

Each traffic source is slightly different. But ultimately, what affects your video’s view count is a mix of:

- **personalization** (the viewer’s history and preferences)
- **performance** (the video’s success)
- **external factors** (the overall audience or market)

Fig 4.7

Chapter 5: Project Literature

When people communicate on a regular basis there is a different level of interaction when we use music to share our feelings. When the user receives a recommendation, why browse through the impersonal music streaming service for the song they're feeling?

A conversation's emotional content determines around 93% of its content. Humans can recognize emotions, which is crucial for effective communication. Business teams may scale their user interactions with the aid of chatbots. Any significant chat application, including Facebook Messenger, Slack, Telegram, and Text Messaging,

services, which improves the user experience. The chatbot would also make song recommendations depending on the user's voice tone. The Last.fm API, a service very similar to Spotify API, will be used to develop the song suggestion feature. The conversation's tone and emotions will also be examined using IBM Tone Analyzer API. Today, API connection is crucial since popular chatbots offer more user-friendly features and engage in much more than simply data-driven conversations. In addition to providing a large selection of open-source libraries for creating chatbots, Python also offers a sizable selection of such libraries.

Fig 5.1
Chatbot adoption across different industries

The diagram accurately depicted how this project will look. Basically, the front end will display the UI, but as soon as a user interacts with the chatbot, the front end will display what they said. Also, utilizing IBM Emotional Intelligence, two apis would be functioning on the server side to identify the user's emotions in their text.

After detecting the emotion, the song will be recommended via the Last.FM API. The user strikes up a conversation the Cakechat Chatbot will respond to your conversation by employing the Last.fm songs API, the app collects the top songs based on the Emotion it senses. The IBM Emotional API is utilized to analyze the conversation's emotional content. Using the Last.fm API, the user will receive recommendations for further music after listening to one song for a while.

With IBM's Tone Analyzer, you may find communication tones in written text. The three different categories of communication tones are social, linguistic, and emotional tones. Anger, contempt, fear, joy, and melancholy are five different emotional tones. Language can be expressed in three different ways: analytically, confidently, and tentatively. The "Big Five" major social characteristics are: openness, conscientiousness, extraversion, agreeableness, and emotional range

are examples of personality qualities. A model was developed to anticipate the tone of fresh texts based on conversations using machine learning. The Support Vector

Machine was used for machine learning (SVM). They chose to employ multi-label classification rather than multi-class classification since about 30% of samples are connected to more than one tone. The One-vs-Rest paradigm was used to train the model separately for each tone. Finding those anticipated with at least a 0.5 probability allowed for the determination of final tones. Shorter sentences cannot be analyzed, so please be aware of that. Text in a single sentence can be up to 128 KB (about 10 sentences). An excellent use case might be customer tweets or Facebook posts on company pages

5.1. Architecture

Fig 5.2
Architecture Diagram

According to Robert Thayer's conventional Emotions model, there are numerous ways to categorize musical moods. The approach divides songs into categories according to their levels of stress and energy, going from happy to sad and from calm to frenetic.

A song with a higher energy level will have a faster tempo, whereas a song with a lower energy level will have a slower tempo. Loud music can convey fury, whereas softer music conveys tenderness, grief, or fear. The tonal elements that harmonics produce determine a piece's mood through its timbre. A louder pitch indicates a higher pitch indicates joy, carefreeness, and lightness, while a lower pitch denotes a serious, melancholy attitude. Researchers from the BNM Institute of Technology in Bangalore, India claim that regardless of rhythm or harmonic saturation, timbre raises people's energy levels. Simple harmonic music typically has a darker timbre and a calming effect on listeners. Songs that are extremely intense symbolize fury, whereas soft tunes convey the "emotions" of compassion, melancholy, and fear. In order to distinguish different moods, they also created a correlation chart of intensity,

timbre, pitch, and rhythm.

Chatbots, also known as conversational interfaces, give humans a new means of interacting with computers. Traditionally, a computer program responded to a query

either by using a search engine or by completing a form. A user can easily ask inquiries through a chatbot in the same way they would through a human. Nowadays, voice chatbots like Alexa and Siri are the most well-known chatbots. Yet, the use of chatbots on computer chat platforms is growing. The primary force behind the creation of chatbots is natural language processing. Chatbots are now a desirable choice for many organizations thanks to recent advancements in machine learning that have improved the accuracy and efficiency of natural language processing. This advancement in NLP is inspiring further study, which should result in ongoing improvements to the chatbot, which has a very promising future as a result of its recent rise to popularity as a website. any programs or software utilized by staff. When using messaging apps to communicate with a big audience, chatbots are more efficient than people. They could be able to gather information in the upcoming years. Natural language processing is used in the current work to create a chatbot with a variety of features and learn about different methodologies.

Fig 5.3
Potential Benefits of Chatbots

5.2. Proposed Model

The suggested system incorporates all three already in-use systems to create a single system that suggests music to users based on the emotional content of their chats with chatbots. Similar songs will start playing after the current song is finished,

or the user can choose to play the next song. This system was created as a combination of an emotive Chatbot, machine learning algorithms to identify the emotional content of communication, and the Spotify music player.

By implementing the API functionality in these modules, we can focus more on the functionality and achieve a simpler system design. API also makes our execution process simpler. A straightforward website is created with a Spotify web player that plays the recommended songs

5.2.1. Algorithm Used

Purpose and scope

The chatbot will be a song recommender, suggesting songs based on user preferences and feedback.

Development platform: You can use a natural language processing (NLP) framework like NLTK or spaCy, along with a machine learning model like scikit-learn, to build the chatbot.

Data collection: Gather data on songs, including title, artist, genre, lyrics, and other features that could be used to identify and recommend songs.

Conversation flow:

User: Hi

Sasha: Hello! How can I help you find some music today?

User: Can you recommend a song for me?

Sasha: Sure, what type of music are you in the mood for? You can tell me a specific genre, artist, or song that you like.

User: I'm in the mood for something upbeat and catchy.

Sasha: Great! I have a few recommendations based on your preferences. Here are three songs you might enjoy: [Blinding Light], [Cold], and [Can't Stop the Feeling]. Let me know if you'd like more information about any of these songs, or if you'd like me to suggest something different.

User: I love Song 2! Can you suggest something similar?

Sasha: Of course! Based on your feedback, here are a few more songs that you might enjoy: [Never really over], [Don't wanna know], and [Hips don't lie]. Let me know if you'd like more information about any of these songs.

Development: Use NLP and machine learning algorithms to process user input, identify relevant song features, and recommend songs based on those features.

Training: Use machine learning algorithms to train the chatbot on the collected data, improving its accuracy and relevancy over time.

Testing and refinement: Test the chatbot with sample user queries and refine its responses based on user feedback.

Deployment: Deploy the chatbot on a messaging platform, website, or mobile app.

Monitoring and maintenance: Monitor the chatbot's performance and make necessary updates and improvements to ensure it remains effective and up-to-date with the latest music trends and user preferences.

Music classification can be difficult because listeners' emotional responses to the same piece can differ. Songs are typically categorized by the artist's overall genre rather than the emotion they arouse. Even though it can be difficult to classify music using engineering methods, it might reduce listeners' differences. Large digital music collections, like those found on iTunes and Spotify, would benefit greatly from the capacity to automatically detect the mood of a composition. Instead of categorizing songs by similar artists, online radio systems like Pandora might utilize mood to determine similar songs. When a song is split down into its quantifiable musical elements like rhythm, harmony, and timbre, it is feasible to match it to particular mood groups into its measurable components.

Fig 5.4

5.3. Natural Language Processing (NLP)

To evaluate whether data is good, negative, or neutral, natural language processing (NLP) techniques like sentiment analysis are applied. Businesses frequently do sentiment analysis on textual data to track the perception of their brands and products in customer reviews and to better understand their target market. To assess whether a passage of text is favorable, negative, or neutral, apply the sentiment analysis process. In text analytics, topics, categories, or entities inside a phrase are

given sentiment scores by combining natural language processing (NLP) and machine learning (ML) approaches.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real-time. There’s a good chance you’ve interacted with NLP in the form of voice operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

5.3.1. *NLP tasks*

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure—these are just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start if those applications are going to be useful.

Several NLP tasks break down the human text and voice data in ways that help the computer make sense of what it's ingesting. Some of these tasks include the following:

- **Speech recognition**, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.
- **Part of speech tagging**, also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies ‘make’ as a verb in ‘I can make a paper plane,’ and as a noun in ‘What make of car do you own?’

- **Word sense disambiguation** is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that determine the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb ‘make’ in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).
- **Named entity recognition**, or NEM, identifies words or phrases as useful entities. NEM identifies ‘Kentucky’ as a location or ‘Fred’ as a man's name.
- **Co-reference resolution** is the task of identifying if and when two words refer

to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., 'she' = 'Mary'), but it can also involve identifying a metaphor or an idiom in the text (e.g., an instance in which 'bear' isn't an animal but a large hairy person).

- **Sentiment analysis** attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from the text.
- **Natural language generation** is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

5.3.2. NLP Use Cases

Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples:

- **Spam detection:** You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more. Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this doesn't match your email experience).
- **Machine translation:** Google Translate is an example of widely available NLP technology at work. Truly useful machine translation involves more than replacing words in one language with words of another. Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language. Machine translation tools are making good progress in terms of accuracy. A great way to test any machine translation tool is to translate text to one language and then back to the original. An oft-cited classic example: Not long ago, translating "*The spirit is willing but the flesh is weak*" from English to Russian and back yielded "*The vodka is good but the meat is rotten.*" Today, the result is "*The spirit desires, but the flesh is weak,*" which isn't perfect, but inspires much more confidence in the English-to-Russian translation.
- **Virtual agents and chatbots:** [Virtual agents](#) such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with the appropriate action or helpful comments. [Chatbots](#) perform the same magic in response to typed text entries. The best of these also learn to recognize contextual

clues about human requests and use them to provide even better responses or options over time. The next enhancement for these applications is a question answering, the ability to respond to our questions—anticipated or not—with relevant and helpful answers in their own words.

- **Social media sentiment analysis:** NLP has become an essential business tool for uncovering hidden data insights from social media channels. Sentiment analysis can analyze the language used in social media posts, responses,

reviews, and more to extract attitudes and emotions in response to products, promotions, and events—information companies can use in product designs, advertising campaigns, and more.

- **Text summarization:** Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read the full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

5.4. IBM Tone Analyzer

Tone Analyzer by IBM helps to detect communication tones in written text. Emotional, linguistic, and social tones are the three types of communication tones. There are five different emotional tones: anger, disgust, fear, joy, and sadness. There are three different linguistic tones: analytical, confident, and tentative. There are five major social traits (Big Five personality traits): openness, conscientiousness, extraversion, agreeableness, and emotional range. What kind of learning has taken place? Using machine learning, a model was created to predict the tone of new texts based on conversations. Machine learning was performed using the Support Vector Machine (SVM). Around 30% of samples are associated with more than one tone, so they decided not to use a multi-class classification, but rather a multi-label classification. The model was trained independently for each tone using the One-vs-Rest paradigm. Final tones were determined by identifying those predicted with at least a 0.5 probability. Please note that shorter sentences cannot be analyzed. A single sentence may contain up to 128KB of text (about 10 sentences). Customer tweets / Facebook posts on company pages would be a good use case.

5.5. Anvil Framework

For the chatbot front end, we are using the ANVIL framework. Anvil is an open source Python-based drag-and-drop web app builder

5.5.1. Anvil's deployment options

Build and deploy your apps in Anvil's cloud – This is by far the easiest way to build a web app. Use our online editor to build your app – then click one button, and it's live on the web!

Build and deploy your apps on your company's private servers – If you want to use Anvil's editor and one-click deployment, but you need it all on your company's private network, check out Anvil Enterprise.

Build in the cloud, host your apps locally – You can build your apps with the Anvil Editor, then check them out with Git and host them on any computer in the world with the Anvil App Server. Read our guide to learn how.

Build an app from scratch with a text editor – There's nothing stopping you! You'll miss out on our drag-and-drop designer and code completion, but there's nothing you can't do. Check out the documentation in our GitHub repository.

5.6. Last.Fm

Last.fm Limited is a music website founded in the [United Kingdom](#) in 2002. Using a music [recommender system](#) called "Audioscrobbler", Last.fm builds a detailed profile of each user's musical taste by recording details of the [tracks](#) the user listens to, either from [Internet radio](#) stations, or the user's computer or [many portable music devices](#). This information is transferred ("scrobbled") to Last.fm's database either via the music player (including, among others, [Spotify](#), [Deezer](#), [Tidal](#), [MusicBee](#), [SoundCloud](#), and [Anghami](#)) or via a [plug-in](#) installed into the user's [music player](#). The data is then displayed on the user's profile page and compiled to create reference pages for individual artists.

5.6.1. Features

User accounts

The free user account includes access to all the main features listed below. Registered Users are also able to send and receive private messages. The newly launched last fm pro user account adds additional features to the free tier, most notable being the ability to change username and early access to new features.

Profile

A Last.fm user can build a musical profile using any or all of several methods: by listening to their personal music collection on a music player application on a computer or an iPod with an Audioscrobbler plugin, or by listening to the Last.fm Internet radio service, either with the Last.fm client, or with the embedded player. All songs played are added to a log from which personal top artist/track bar charts and musical recommendations are calculated. This automatic track logging is called scrobbling.

Last.fm automatically generates a profile page for every user which includes basic information such as their user name, avatar, date of registration and the total number of tracks played. There is also a Shoutbox for public messages. Profile pages are visible to all, together with a list of top artists and tracks, and the 10 most recently played tracks (can be expanded). Each user's profile has a 'Taste-o-Meter' which gives a rating of how compatible the user's music taste is.

Recommendations

Last.fm features a personal recommendations page that is only visible to the user concerned and lists suggested new music and events, all tailored to the user's own preferences. Recommendations are calculated using a collaborative filtering algorithm so users can browse and hear previews of a list of artists not listed on their own profiles but which appear on those of others with similar musical tastes.

Music Classification Based on Mood

Classifying music can be challenging because listeners' emotional reactions can vary for the same song. Generally, songs are classified by the overall genre of the artist rather than the feeling evoked by the song. Despite the challenges of categorizing music using engineering techniques, it may help to minimize these discrepancies between listeners. The ability to identify the mood of a piece automatically would be extremely useful for sorting large collections of digital

music, such as those found on iTunes and Spotify. Online radio services like Pandora could also use mood to determine similar songs, rather than identifying them by similar artists. It is possible to match songs to specific mood categories based on quantifiable musical components like rhythm, harmony, and timbre when the song is broken down into its quantifiable elements.

There are a number of existing methods of dividing moods in music according to Robert Thayer's traditional model of Feelings. The model consists of dividing songs based on their levels of energy and stress, from happy to sad and calm to energetic, respectively (Bhat et al 359). Using Thayer's Model he created eight categories that included the extremes of the lines as well as the points where the

lines may cross each other (e.g. happy-energetic or sad-calm). Figure 1 illustrates the model.

Fig 5.5

A faster tempo corresponds to a song that has high energy, while a slower tempo corresponds with a song with lower energy. Songs that are loud can be associated with anger, while softer songs suggest tenderness, sadness, or fear. Harmonics create tonal components of a piece that inform the mood of a piece by their timbre. A higher pitch means happiness, carefree mood, and light mood, while a lower pitch means serious, mournful mood. According to researchers at the BNM Institute of Technology in Bangalore, India, timbre stimulates the human energy levels regardless of rhythm or harmonic saturation. Music with simple harmonic profiles tends to have a darker timbre and tends to soothe people. Songs with a lot of intensity represent anger, while soft songs express tenderness, sadness, and fear “emotions”. They also developed a correlation table of intensity, timbre, pitch, and rhythm in order to identify various moods.

5.7. Survey

Conversational interfaces (also known as chat bots) present a new way for individuals to interact with computers. Traditionally, a software program answered a question by using a search engine or by filling out a form. With chat bot, a user can simply ask questions in the same way they would with a human. Currently, the most well-known chatbots are voice chat bots, such as Alexa and Siri. Nevertheless, chat bots are becoming increasingly popular on computer chat platforms. Natural language processing is at the core of the rise of chatbots. A recent improvement in machine learning has made natural language processing

more accurate and effective, making chatbots an attractive option for many organisations. This improvement in Natural Language Processing is motivating additional research, which should lead to continuous improvements in the Chatbot, which has a very bright future since it will become very common as a website in recent years. Additionally, it is not that expensive, so anyone who has a website can afford it. In recent years, chat bots have become increasingly prevalent in society. There are a lot of studies on chat bots that use different algorithms and how to create advanced chat bots. This study relies heavily on expert knowledge. Any software or applications used by personnel. Chatbots are more effective than humans at reaching out to a large audience via messaging apps. Within the next few years, they may be capable of gathering information. Using natural language processing, the present research seeks to develop a chat bot with various features and to learn about different algorithms.

5.8. Tools and Technologies

5.8.1. Python

Python is a widely used high-level programming language for general-purpose programming. Python has an automatic memory management system and a dynamic type system. Multiple programming paradigms are supported, including object-oriented, imperative, and functional. Styles of programming and procedural programming. The standard library is extensive and comprehensive.

5.8.2. TensorFlow

TensorFlow is an infrastructure layer that enables differentiable programming. It's a framework for manipulating N-dimensional arrays (tensors), similar to NumPy. However, there are three key differences between NumPy and TensorFlow: TensorFlow leverages hardware accelerators such as GPUs and TPUs.

TensorFlow can compute gradients of arbitrary differentiable tensor expressions. It is possible to distribute TensorFlow computation to large numbers of devices on a single machine and to large numbers of machines (possibly with multiple devices per machine).

5.8.3. Keras layers

Keras is a user interface for deep learning, handling layers, models, optimizers,

loss functions, metrics, and more. TensorFlow handles tensors, variables, and gradients, which are part of differentiable programming.

TensorFlow's Keras API is what makes it simple and productive: Keras is what makes TensorFlow easy and simple. In Keras, the Layer class represents the fundamental abstraction. Layers encapsulate a state (weights) and some computation (defined in the call method).

IBM Analyser Conduct Social Listening: Analyze emotions and tones in what people write online, such as tweets and reviews.

Find out whether they are happy, sad, confident, etc.

Enhance Customer Service: You can respond to your customers effectively and at scale by monitoring customer service and support conversations. Verify

whether customers are satisfied or frustrated, and whether agents are polite and sympathetic. Chatbots should be able to distinguish customer tones so they can build dialog strategies that adjust the conversation accordingly.

Chapter 6: Conclusion and Future work

6.1. Conclusion

This project focuses on a combination of three different technologies – Chatbot, Emotional Tone Analysis, and Song Recommendation systems with our sub-focus of the project on API and Flask framework implementations. It also covers the domain of ML and AI, how these concepts are integrated with a chatbot, and how it affects the responsiveness of the chatbot. The emotional analysis when combined with song recommendation can understand users' emotions and play songs that are relevant to users' emotions. The project was successful at getting the user response from the user, analysing the emotional tone of the user, and displaying the songs to the user and the songs displayed show similar recommendations to the user. This project will be useful for songs providing platforms such as Spotify, SoundCloud, iTunes, YouTube Music, and more for better understanding their user base and delivering optimal songs according to the mood of the user which will in turn enhance the user experience

Fig 6.1

6.2. Future use

From assessing purchase intent to answering computer-related questions, chatbots

46

are well on their way to playing a major role in contemporary business. Chatbots are fully functional semi-stand-alone systems that can help with customer service experiences and response time.

However, this does not mean that their future in the company is safe. For chatbots to withstand the rapidly-increasing technological changes and become pillars of the business, developers must look at the problems that have arisen with increased implementation.

6.2.1. Future bots

Although the voice interface can be optional, chatbots have been in the business for

quite some time for developers and experts to begin identifying which components of chatbots are maintenance requirements. The development of NLP, conversational flexibility and 24/7 service are critical to maintaining the longevity of chatbots in corporate environments. Chatbots are AI devices and, going forward, they must follow AI trends, such as machine-based learning, easy system integration, and intelligence development.

Chatbots must have a basis for natural language processing, learning and comprehension. This extends to the interpretation of the user's intention through the development of a domain-specific language and enhance their functionality to adapt and modify them to suit the specifics of any conversation. As the chatbot novelty dissipates and users become more demanding of operational capability, scalability will become more and more important.

The latest trends in chatbot are an excellent example. It is possible to improve the user experience by using chatbots to make things easier, simpler, and more streamlined. Greater interoperability allows different technologies to work together smoothly and efficiently, offering users a better overall experience. The development of standards such as JSON API is paving the way for more developers to get involved with creating bots that communicate across multiple channels. By setting APIs as common building blocks between systems, new possibilities arise in terms of suggestions, alerts, and actions that virtual assistants may take on behalf of their users making use of the interoperability benefits.

One of the most notable challenges is developing natural language processing (NLP) capabilities for the chatbot. This is critical for a chatbot to be able to understand requests from users and provide meaningful responses. Additionally, NLP technologies are necessary for the chatbot to be able to identify topics related to the user's query or request.

Another major challenge faced by the chatbot industry is developing a wide range of knowledge bases. Knowledge bases are collections of information that the chatbot can access to provide answers to user questions. As the number and variety of potential inquiries grow, so must the knowledge base. This is one of the biggest hurdles for developers looking to use chatbots for more advanced tasks such as customer service and product recommendations.

47

Finally, developing an effective user interface is essential for a successful chatbot. Chatbots must be designed so that users can easily interact with them and understand their responses. This involves designing a chatbot with the right tone of voice and personality, as well as developing graphical user interfaces (GUIs) that make it easy for users to navigate and access information.

We use voice assistants almost regularly for navigation and typing messages on our phones. And enterprises are starting to leverage voice assist solutions as part of their user experience strategy. With conversational commerce, customers can communicate easily with online and in-store brands.

Websites require more than SEO-friendly content; they need to be personalized to encourage users to buy more often. A chatbot can help promote online business

by providing users with a personalized buying experience to communicate and purchase remotely without visiting a physical store.

Our poll results show that more than 63 percent of businesses agree that having chatbots on their website increases conversion rates. These chatbots are designed to be simple and encourage customers to progress through the conversion funnel. Chatbots are constantly evolving and becoming more widely used to offer the best possible customer service.

By interacting with customers in real-time, customer service chatbots can provide a level of service that is otherwise not possible. By the end of 2021, chatbots will have helped businesses gain market share and be a significant investment in customer service experience.

If the future demands advanced chatbots that do more than use scripted, single turn exchanges, then their method of interface will also have to advance. A voice interface can assist users with disabilities or those who are skeptical of technology, but it also requires another layer of NLP development.

While some experts such as Ramesh Hariharan, co-founder and CTO of LatentView Analytics, say that using a voice interface is foundational to the future success of chatbots, other developers argue for a separation between fully functioning chatbots and voice-based digital assistants.

Wai Wong, founder and CEO of Serviceaide, a California-based software company, advised developers to approach voice interface on a case-by-case basis. The future is not all voice interfaces, but instead it should focus on working within device capabilities to boost user accessibility and flexibility. Some chatbots can offer voice-to-text interfaces, voice-to-voice interfaces or text-to-voice interfaces, all depending on customer need or brand decision.

"The important question is how a conversational platform interacts with the voice capability already existing: supplementing what is there or integrating into the on device capability," Wong said.

References

1.K. S. Naveen Kumar, R. Vinay Kumar and K. P. Soman, "Amrita-CEN-SentiDB 1: Improved Twitter Dataset for Sentimental Analysis and Application of Deep learning", 2019 10th International Research on Computing Communication and Networking Technologies (ICCCNT), pp. 1-5, 2019.

2.H.Immanuel James, J.James Anto Arnold, J.Maria Masilla Ruban, M. Tamilarasan and R. Saranya, "Emotion based Music Recommendation System", International Research Journal of Engineering and Technology(IRJET), vol. 06, no. 03, Mar 2019.

3. K. Jwala is pursuing M. Tech.(CST) in the department of CSE in S.R.K.R Engineering College, India. She did her B. Tech.(I.T) in the same college. This is the first paper that is going to be published by her.
4. Dr. G. V. Padma Raju is a professor and Head of the department of CSE in S.R.K.R. Engineering College, India. He did his B. Sc.(Physics) from Andhra University and stood second in M. Sc.(Physics) from IIT, Bombay. He stood first in university level in M. Phil. (Computer Methods). He received Ph.D. from Andhra University. He has 30 years working experience as a lecturer
5. Peter Bae Brandtzæg and Asbjørn Følstad. 2018. Chatbots: Changing User Needs and Motivations. *Interactions* 25, 5 (2018), 38–43.
6. N. Dahlback, A. Jönsson, and L. Ahrenberg. 1993. Wizard of Oz studies — why and how. *Knowledge-Based Systems* 6, 4 (1993), 258 – 266.
7. Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. 2005. Wizard of Oz Support Throughout an Iterative Design Process. *IEEE Pervasive Computing* 4, 4 (Oct. 2005), 18–26