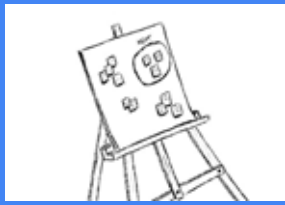




Chatbots Introduction

Ailton Moreira
ailton.moreira@openup.pt

Lisboa 2024



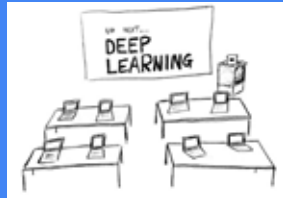
Consulting



Support



Services



Training



We Work With Data

We Work With
Big Data & Data Science

We are Data
Scientists



ailton.moreira@openup.pt

- PhD candidate in Technology and Information Systems;
- Invited Assistant Professor at Uminho;
- Researcher and Information Systems Engineer at ALGORITMI Center;
- Full stack developer;
- Work in different Data Science projects.

ailton.moreira@openup.pt

- Briefly introduce yourself:
 - Name
 - Title or position
 - Company
 - Experience with chatbots or python
 - Expectations

- Introduction to the World of Chatbots
 - What is a Chatbot?
 - Why are Chatbots Important?
 - Practical Applications of Chatbots in the Business World
- Getting to know Chatbot Platforms
 - Popular Tools and Platforms
 - Pros and Cons of Each Platform
- Getting Started: Creating Your First Chatbot
 - Choosing the Ideal Platform
 - Setting Up Your First Chatbot
 - Understanding Intents, Entities and Dialogues

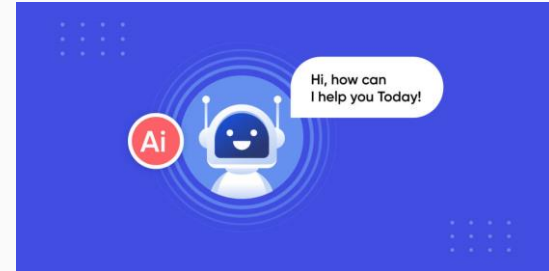


- Making Your Chatbot Smart
 - Integrating with Artificial Intelligence
 - Training and Machine Learning
 - Natural Language Recognition
- Conversational Design
 - Principles of a Good Dialogue
 - Keeping Users Interested
 - Handling Errors and Complex Questions
- Integrating Chatbot in Diverse Channels
 - Websites and Mobile Applications
 - Social Media Networks
 - Customer Service Systems

- Testing and Optimizations
 - How to Test Your Chatbot
 - Collecting User Feedback
 - Improving and Expanded Functionalities
- Conclusion and Next Steps
 - Future Challenges in the World of Chatbots
 - Trends and New Technologies
 - How to Stay Updated

A chatbot is a software application that simulates human conversation through text or voice interactions. The concept of chatbots dates back to the 1960s with the creation of ELIZA, a computer program that simulated conversation. Since then, advancements in artificial intelligence and natural language processing have significantly enhanced the capabilities of chatbots.

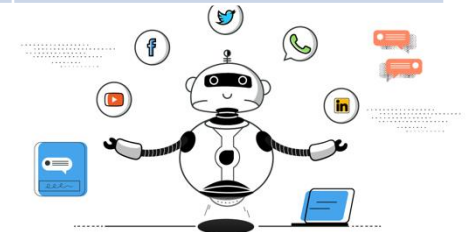
These days, chatbots are starting to integrate conversational AI, such as natural language processing (NLP), to understand questions even if it isn't grammatically correct and then respond based on data they have collected.



Chatbots vs. AI chatbots vs. virtual agents vs. virtual assistants



	Chatbot	AI Chatbot	Virtual Agent
What is it?	Most inclusive. Any software that simulates human conversation	Chatbot that uses AI technology, from machine learning to NLP	AI chatbot that adds on robotic process automation to act upon user desires without prompting
Where are they often found?	Any communication channel, from social media to websites	On your phone or home	In specific customer service instances
Example	Customer service chatbot for health care company	Alexa and Siri, Google Assistant, ChatGPT	IBM watsonx Assistant



- Chatbots process collected data and often are trained on that data using AI and machine learning (ML), NLP, and rules defined by the developer. This allows the chatbot to provide accurate and efficient responses to all requests. The two main types of chatbots are declarative chatbots and predictive chatbots.
- Chatbot type is determined by its capabilities. Declarative chatbots are more basic than predictive chatbots.

- Declarative chatbots perform one function. These chatbots use NLP, defined rules, and ML to generate automated responses when you ask a question.
- Declarative, or task-oriented chatbots, are most common in customer support and service—and are best when answering commonly asked questions like what the store hours are and what item you're returning.
- This type of chatbot is common, but its capabilities are basic compared to predictive chatbots.
- Rule-based Chatbots - They operate on predefined rules and patterns. They follow a set of guidelines programmed by developers to provide specific responses to specific inputs.

- Predictive chatbots are more sophisticated and personalized than declarative chatbots.
- Often considered conversational chatbots, or virtual agents, these AI- and data-driven chatbots are much more interactive and aware.
- They utilize NLP and more complicated ML, along with natural language understanding (NLU) to continue learning about the user through predictive analytics and intelligence.
- Over time, they can even predict recommendations and anticipate your needs.

Chatbots are important because they are a valuable extension of your support team, helping both customers and employees. Key benefits of chatbots vary from 24/7 support to personalized conversations.

- **24/7 support:** Customers today expect help as soon as they need it on the channels of their choice. Deploying a chatbot across preferred customer channels ensures customers get seamless, always-on support.
- **Scalability:** Small business and enterprise chatbots can serve as extra support agents, answering FAQs without human intervention. As a result, teams can scale quickly and assist more customers without hiring more staff.

- **Conversion opportunities:** Customer service bots can boost lead conversations, nudging consumers to take action by presenting them with image carousels, forms, picklists, and other messaging elements. This also helps increase customer engagement.
- **Personalized conversations:** As the first point of contact, chatbots help agents by capturing customer details like name, issue type, and contact information. This enables the agent to personalize the conversation when they speak with the customer.

- Real-world Use Cases

Chatbots are used in various industries, including:

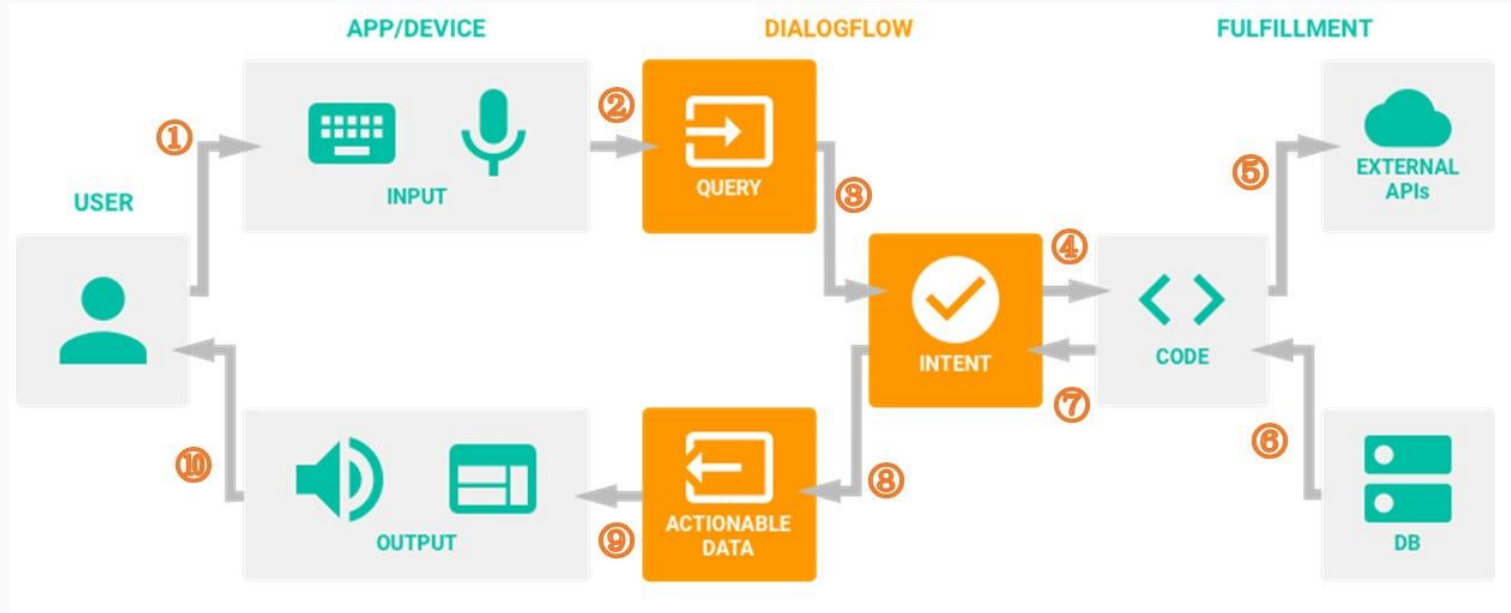
- **Customer Service:** Answering FAQs, handling complaints, and guiding users through processes.
- **E-commerce:** Assisting with product recommendations, order tracking, and transaction processing.
- **Healthcare:** Providing medical information, scheduling appointments, and sending reminders.
- **Finance:** Managing account queries, processing payments, and offering financial advice.

Overview of Popular Platforms

Several platforms are available for developing chatbots, each with its strengths:

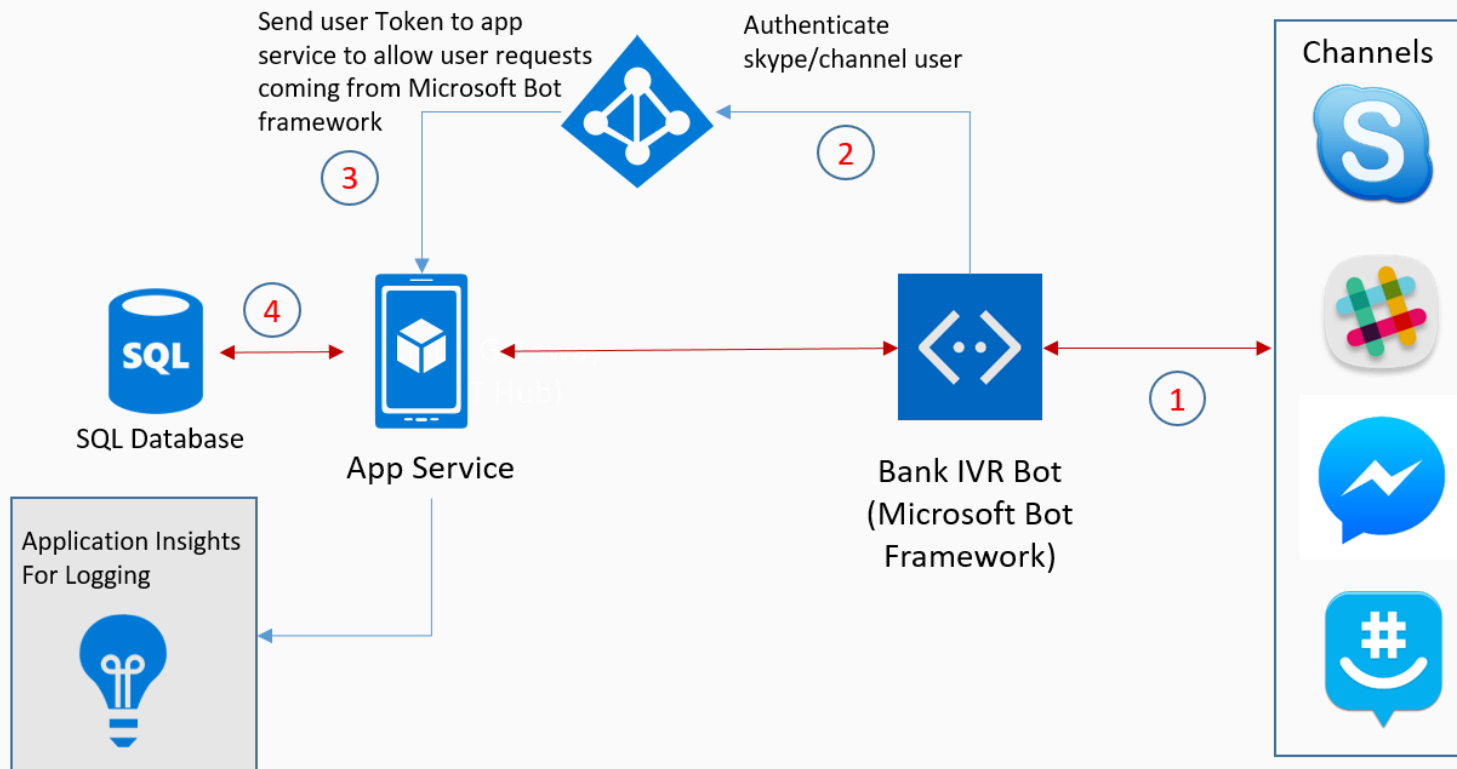
- **Dialogflow:** Developed by Google, it supports voice and text-based conversational interfaces.
- **Microsoft Bot Framework:** A comprehensive platform for building and connecting bots across multiple channels.
- **Rasa:** An open-source machine learning framework for building contextual AI assistants.
- **ChatterBot:** A Python library that makes it easy to generate automated responses to user inputs.

Dialogflow

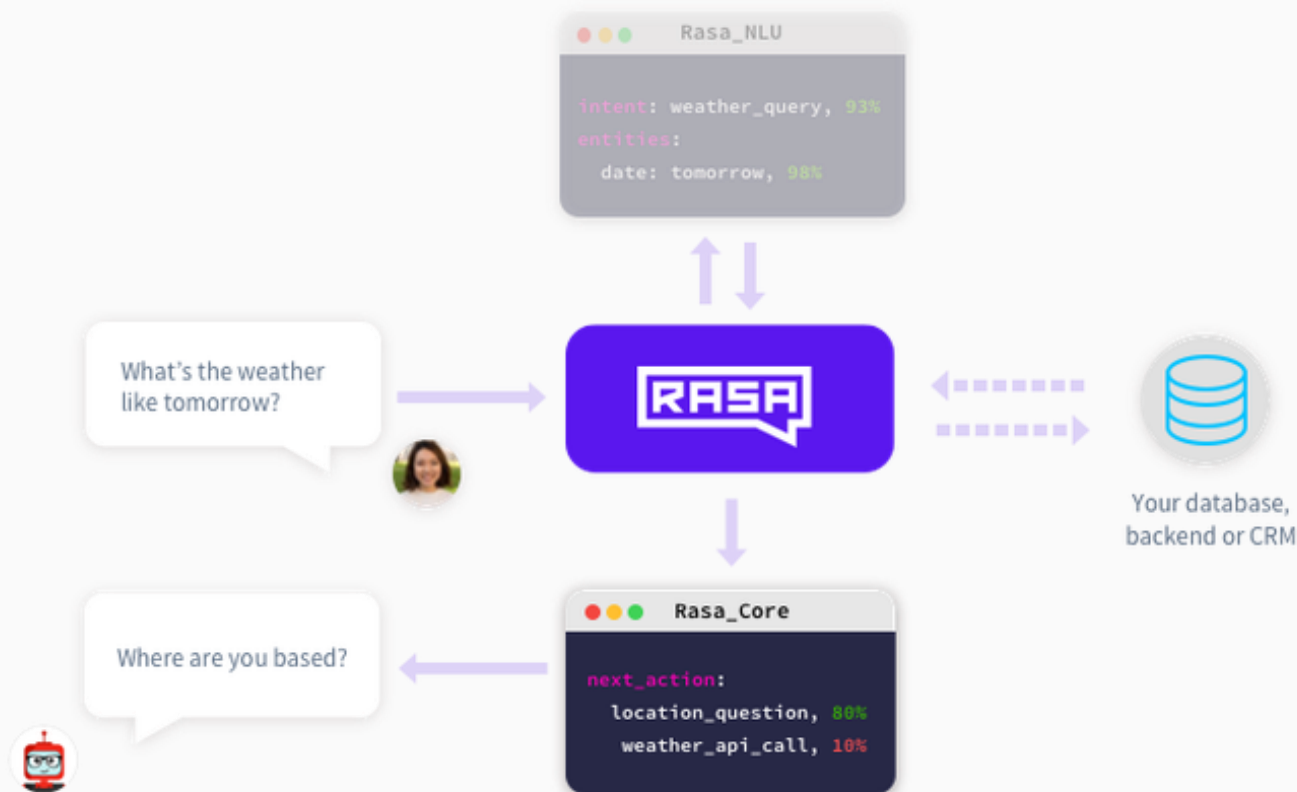


[Demo](#)

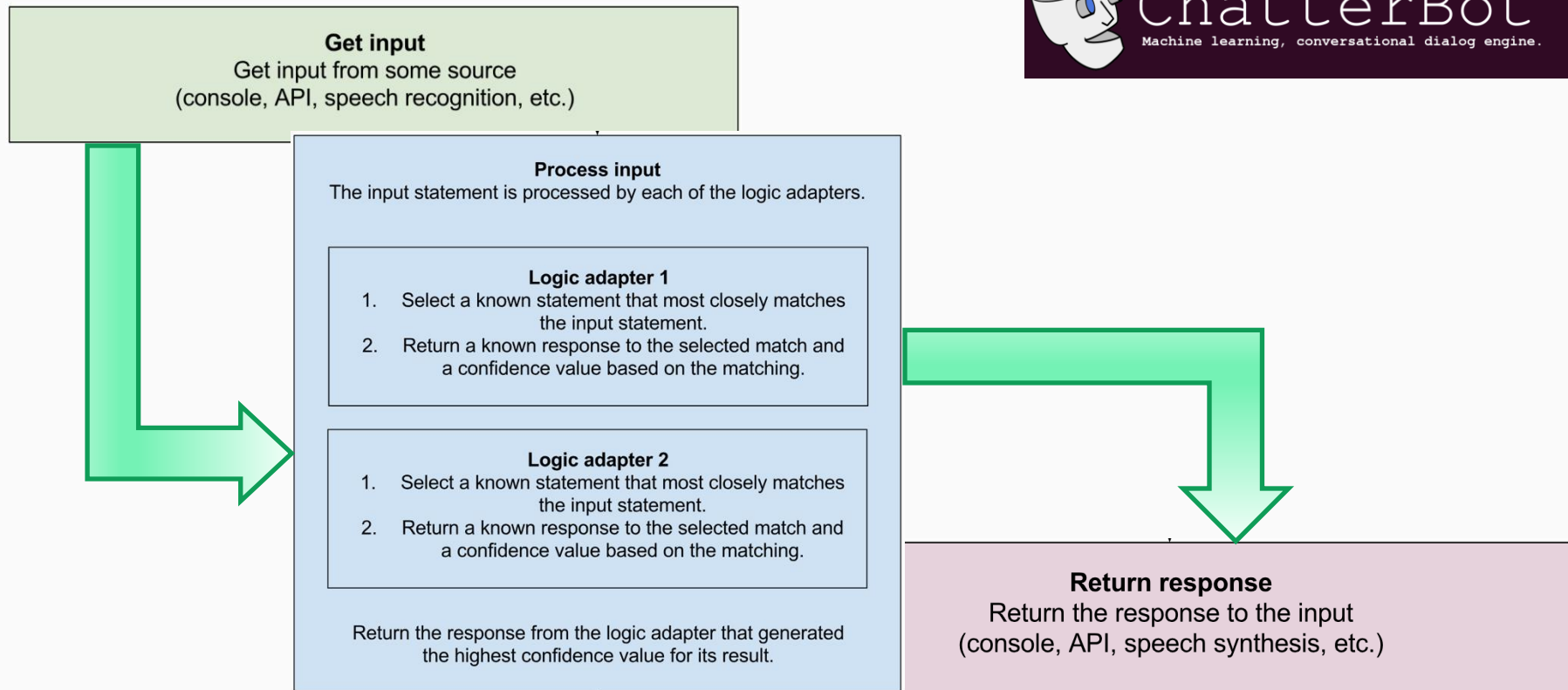
Microsoft Bot Framework



Rasa



ChatterBot



	Dialogflow	Microsoft Bot Framework	Rasa	ChatterBot
Pros	Easy to use, powerful NLP, supports multiple languages.	Extensive integration options, robust development tools	Highly customizable, open-source, great for complex scenarios	Simple setup, good for beginners
Cons	Dependency on Google services, limited customization	Can be complex to set up, higher learning curve	Requires more development effort, less beginner-friendly	Basic functionality, limited scalability

Selecting the best chatbot platform for your business is not as easy as it seems.

- **Determine your aim and capabilities**
- **Integrations**
- **AI capabilities**
- **Pricing**
 - **Limits**
 - **Key features**
 - **Scalability**
 - **Hidden costs**

- Choosing the Ideal Platform

When choosing a platform, consider the following:

- **Complexity of the chatbot:** For simple interactions, **ChatterBot** is a good start (Python library). For more advanced requirements, consider Dialogflow or Rasa.
- **Integration needs:** Microsoft Bot Framework excels in integrating with various services.
- **Customization and control:** Rasa offers the highest level of customization and control.

- **Step-by-Step Guide**
- **Install Python and Libraries:** Ensure you have Python installed. Set up your development environment by installing the necessary libraries.
- **Select a Platform:** For this course, we'll start with [ChatterBot](#) due to its simplicity.
- **Basic Setup and Configuration:** Follow the installation instructions and configure your first chatbot.
- **Understand Intents, Entities, and Dialogues:** Learn the basic concepts of how chatbots understand and process user inputs.

- **Key Concepts**
- **Intents:** Represent the purpose of a user's input (e.g., greeting, booking a flight).
- **Entities:** Extract specific pieces of information from user inputs (e.g., dates, locations).
- **Dialogues:** Manage the flow of conversation, ensuring the chatbot responds appropriately to user inputs.

- **Intents**

Intents represent the purpose or goal behind a user's input. They are essentially the user's intention for interacting with the chatbot. For example, in a customer service chatbot, typical intents might include:

- Greeting: When the user says "Hello" or "Hi".
- Booking: When the user wants to book a ticket, appointment, etc.
- Information Request: When the user asks for information about a product or service.
- Goodbye: When the user says "Goodbye" or "Bye".

- **Entities**

Are specific pieces of information that can be extracted from the user's input, providing more context and detail about the intent. They are the data points that the chatbot needs to fulfil the user's request. For example:

- In the intent "Book a flight," entities might include:
- Destination: The city or country to which the user wants to travel (e.g., "Amsterdam").
- Date: The date on which the user wants to travel (e.g., "September 10th").
- Class: The travel class (e.g., "economy", "business").

- **Dialogues**

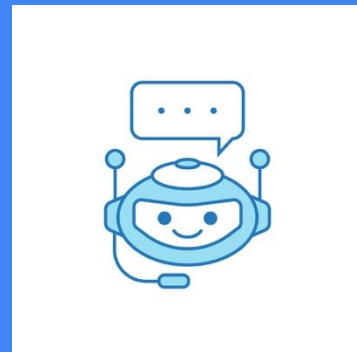
Dialogues are the flow of conversation between the user and the chatbot. They involve managing the sequence of interactions, ensuring that the chatbot responds appropriately at each step. Dialogues encompass:

- Prompting the user for information: Asking follow-up questions to gather necessary details.
- Handling user responses: Processing and validating the information provided by the user.
- Executing actions: Performing tasks based on the user's input and entities (e.g., booking a flight, providing information).
- Providing feedback: Confirming actions or giving relevant responses to the user.



We Work With Data

Thank you!



ailton.moreira@openup.pt



Chatbots Introduction

Ailton Moreira
ailton.moreira@openup.pt

Lisboa 2024

- Making Your Chatbot Smart
 - Integrating with Artificial Intelligence
 - Training and Machine Learning
 - Natural Language Recognition
- Conversational Design
 - Principles of a Good Dialogue
 - Keeping Users Interested
 - Handling Errors and Complex Questions
- Integrating Chatbot in Diverse Channels
 - Websites and Mobile Applications
 - Social Media Networks
 - Customer Service Systems



- Integrating with Artificial Intelligence

- Incorporate NLP for Understanding and Generating Language

- SpaCy: An NLP library that can handle tasks like tokenization, part-of-speech tagging, named entity recognition, etc.
 - NLTK: Another library for NLP, useful for text processing and linguistics.
 - Transformers: Models from Hugging Face (e.g., BERT, GPT-3, GPT- 3.5 turbo, GPT-4o) for sophisticated language understanding.



- Integrating with Artificial Intelligence

- Use Pre-trained Models

- Integrate pre-trained models like GPT-3, and GPT- 4o, which can understand and generate human-like text (extensive API docs).

- Train on Relevant Data

- Collect and clean a dataset specific to your domain.
 - Fine-tune models on this dataset to improve their relevance and accuracy.



- Integrating with Artificial Intelligence

- Implement Dialogue Management

- Use frameworks like Rasa to manage the flow of conversation, handle user intents, and maintain context.

- Add Machine Learning techniques

- Train custom models to classify intents, detect sentiment, and provide appropriate responses.



- Integrating with Artificial Intelligence

Advantages

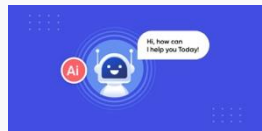
- Capabilities to use the knowledge base from already trained LLM
- Readily Available
- Better Engagement
- Enhanced Data Collection
- Reduced Initial Expenses
- Personalization



- Integrating with Artificial Intelligence

Disadvantages

- May Pose a Data Security Threat
- May Be Unable to Understand Human Emotions
- May Be Inaccurate
- Hidden Cost Associate
- Complexity



- Integrating with Artificial Intelligence

```
# Integrate GPT-3 for Advanced Responses:
import openai # type: ignore

openai.api_key = 'YOUR_GPT3_API_KEY'

def gpt3_response(prompt):
    response = openai.Completion.create(
        engine="text-davinci-003",
        prompt=prompt,
        max_tokens=150
    )
    return response.choices[0].text.strip()
```



- Training and Machine Learning

- Data Collection and Preparation

- Collect a dataset of conversations relevant to the chatbot's domain. This could include transcripts from customer support, chat logs, etc.
 - Clean and preprocess the data to ensure it's in a format suitable for training. This includes tasks like tokenization, removing stopwords, and normalizing text.

- Feature Engineering

- Extract features from the text data. Common features include n-grams, TF-IDF (term frequency-inverse document frequency), and word embeddings (e.g., using Word2Vec, GloVe, or BERT).



- Training and Machine Learning

- Labelling Data

- If the training model is for classification intent, the data must be labelled with the correct intents.
 - For training a response generation model, ensure that each input is paired with an appropriate output response.

- Choosing Machine Learning Models

- Use algorithms like Support Vector Machines (SVM), Random Forest, or deep learning models like RNNs, LSTMs, or transformers for intent classification.
 - Use sequence-to-sequence models, Transformers (e.g., GPT, BERT), or encoder-decoder architectures for response generation.



- Training and Machine Learning

- Training the Model:

- Split the dataset into training and testing sets to evaluate your model's performance.
 - Train the model using the training data and validate it using the testing data. Use appropriate metrics (e.g., accuracy, precision, recall) to measure performance.

- Fine-Tuning:

- Fine-tune hyperparameters to improve model performance. This can include adjusting learning rates, batch sizes, or the number of layers/units in your neural network.



- Training and Machine Learning

- Integration

- Integrate the trained model into the chatbot framework. This can be done by calling the model's prediction functions to classify intents or generate responses during a conversation.

- Continuous Learning

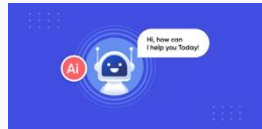
- Implement mechanisms for the chatbot to learn continuously from new data. This can involve periodically retraining the model with new data or using online learning techniques.



- Training and Machine Learning

Advantages

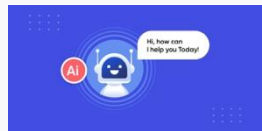
- Improved Accuracy
- Adaptability
- Personalization



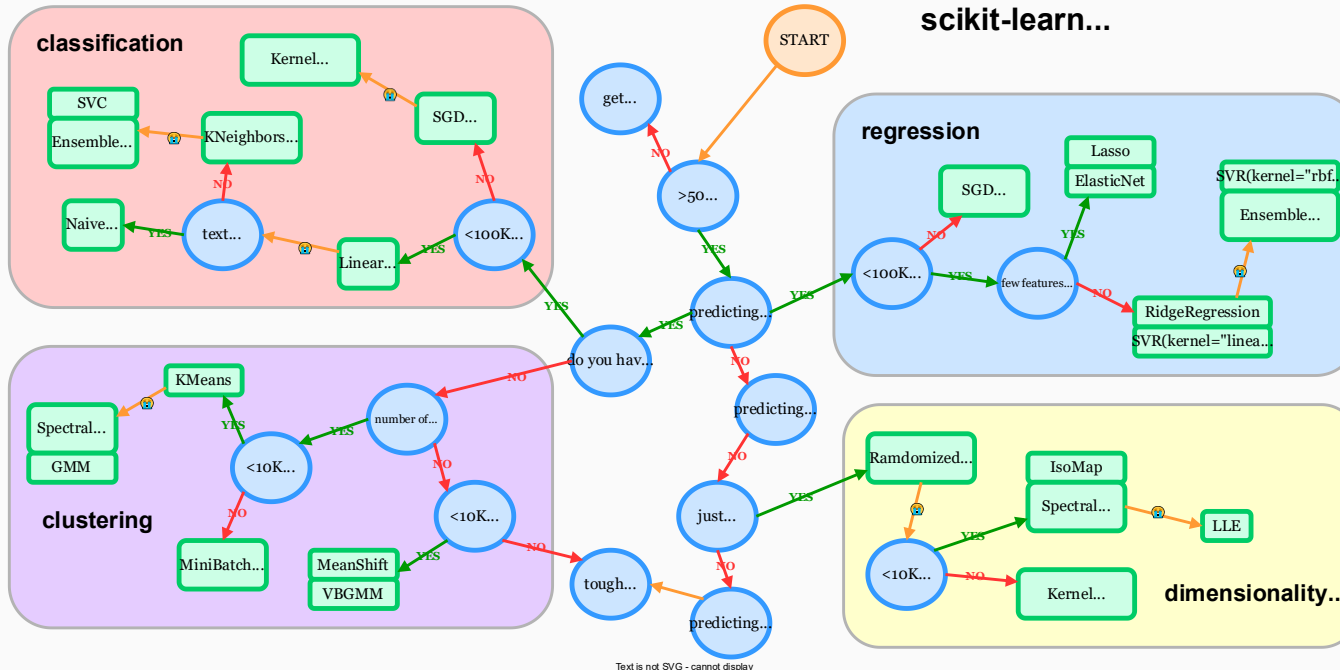
- Training and Machine Learning

Disadvantages

- Data Dependency
- Complexity
- Resource Intensive



- Training and Machine Learning



- Natural Language Recognition

- Intent Recognition:

- Recognize the user's intention behind the input (e.g., greeting, asking for help, querying order status).
 - Libraries: Rasa, SpaCy, NLTK, Transformer models (e.g., BERT, GPT-3).

- Entity Extraction:

- Identify and extract specific pieces of information from the text (e.g., dates, names, order numbers).
 - Libraries: SpaCy, Rasa, Stanford NLP.



- Natural Language Recognition

- Sentiment Analysis:

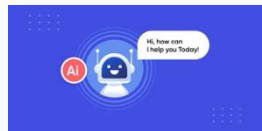
- Determine the sentiment or emotion conveyed in the user input (e.g., happy, frustrated).
 - Libraries: TextBlob, VADER (part of NLTK), Hugging Face Transformers.



- Natural Language Recognition

Advantages

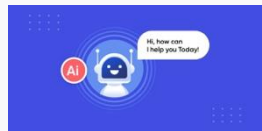
- Better Understanding
- Context Awareness
- Efficiency



- Natural Language Recognition

Disadvantages

- Data Requirement
- Complexity
- Maintenance



- Natural Language Recognition

```
### Natural Language Recognition
import spacy # type: ignore

nlp = spacy.load("en_core_web_sm")

def extract_entities(text):
    doc = nlp(text)
    entities = [(ent.text, ent.label_) for ent in doc.ents]
    return entities

text = "My order number is 12345 and I placed it on January 10th."
print(extract_entities(text))
# Output: [('12345', 'CARDINAL'), ('January 10th', 'DATE')]
```



Conversational design involves creating dialogues that are intuitive, engaging, and effective.

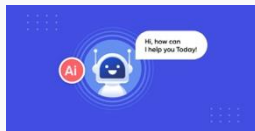
- Principles of Good Dialogue

- Clarity and Simplicity

- Use clear and simple language to avoid confusion.
 - Ensure that the chatbot's responses are easy to understand.

- Consistency

- Maintain a consistent tone and style throughout the conversation.
 - Follow established conventions for user interactions.



- Principles of Good Dialogue

- Context Awareness

- Keep track of the conversation context to provide relevant responses.
 - Avoid asking for information that the user has already provided.

- User Guidance

- Guide the user through the conversation with prompts and suggestions.
 - Offer clear options and next steps to keep the conversation flowing.



- Principles of Good Dialogue

- Personalization

- Personalize responses based on user data and previous interactions.
 - Use the user's name and reference past conversations where appropriate.

- Handling Errors

- Handle misunderstandings and errors gracefully.
 - Provide helpful feedback and guide the user back on track.



- Principles of Good Dialogue

- Brevity

- Keep responses concise and to the point.
 - Avoid overwhelming the user with too much information at once.

- Engagement

- Engage users with a friendly and approachable tone.
 - Use humour and personality where appropriate to make the interaction enjoyable.



- Keeping Users Interested

- Personalization

- Tailor responses to individual users based on their preferences, history, and behaviour.
 - Use the user's name and reference past interactions to make the conversation feel more personal.

- Interactive Elements

- Use buttons, quick replies, carousels, and other interactive elements to make the conversation dynamic.
 - Provide visual elements like images, videos, and GIFs to enhance engagement.



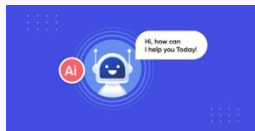
- Keeping Users Interested

- Timely and Relevant Information

- Provide timely updates and information that are relevant to the user.
 - Use real-time data to keep the conversation current and valuable.

- Conversational Tone

- Maintain a friendly, approachable, and sometimes humorous tone to make the chatbot more relatable.
 - Avoid overly formal or robotic language.



- Keeping Users Interested

- Gamification

- Incorporate gamified elements like quizzes, challenges, and rewards to make the interaction fun.
 - Offer points, badges, or other incentives for engaging with the chatbot.

- Providing Value

- Ensure that each interaction provides some value to the user, whether it's answering a question, solving a problem, or providing useful information.
 - Avoid repetitive and irrelevant responses.



- Keeping Users Interested

- Storytelling

- Use storytelling techniques to make the interaction more engaging and memorable.
 - Guide users through a narrative that makes the conversation flow naturally.



- Handling Errors

- Graceful Degradation

- When the chatbot encounters an error, it should fail gracefully by providing a helpful response and guiding the user back on track.
 - Use fallback responses to handle unrecognized inputs or system errors.

- Clarifying Questions

- If the chatbot does not understand a user's query, it should ask clarifying questions to get more context.
 - Use context-aware prompts to gather the necessary information.



- Handling Errors

- User Feedback Loop

- Allow users to provide feedback on incorrect responses.
 - Use this feedback to improve the chatbot's understanding and responses over time.

- Predefined Error Responses

- Have a set of predefined responses for common errors or unknown intents.
 - These responses should be informative and guide the user on what to do next.



- Handling Complex Questions

- Decomposing Complex Questions

- Break down complex queries into smaller, manageable parts.
 - Handle each part separately to construct a comprehensive response.

- Context Management

- Maintain the context of the conversation to handle multi-turn dialogues effectively.
 - Use context to keep track of previous interactions and reference them as needed.



- Handling Complex Questions

- Fallback Mechanisms

- Implement fallback mechanisms for questions that the chatbot cannot handle directly (security questions).
 - Provide options for escalating to human support or accessing external resources.

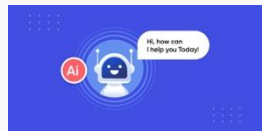
- Utilizing Knowledge Bases

- Integrate the chatbot with a knowledge base or FAQ system to answer complex questions.
 - Use semantic search to retrieve relevant information from the knowledge base.



- Websites and Mobile Applications

- Create an API for the Chatbot
 - Flask
 - FastAPI
- Create a REST API and deploy it to the web server or cloud server, ensuring it is accessible over the internet.
- Integrate the API with web applications using AJAX, Axios request or WebSockets.
- For mobile, it can be used platform-specific HTTP libraries to make a request to the chatbot API.



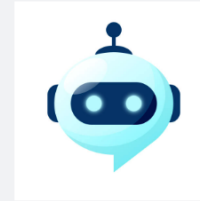
- Websites and Mobile Applications

- Flask

```
app.py > ...
1  from flask import Flask, request, jsonify, render_template # type: ignore
2  from chat import get_response
3  app = Flask(__name__)
4
5  @app.route('/')
6  def home():
7      return render_template("index.html")
8
9  @app.route('/get')
10 def get_bot_response():
11     userText = request.args.get('msg')
12     response = get_response(userText)
13     return response
14
15 if __name__ == '__main__':
16     app.run()
```

Python ChatBot with AI, ML

Please start your personalized interaction with the chatbot



Hi! I'm your Python Chatbot

hey

Hello! What brings you here today?

tell me a joke

Why don't some couples go to the gym? Because some relationships don't work out.

Message



- Social Media Networks

- Meta developer API
- X developer API
- Examples of open-source projects
 - python-telegram-bot
 - Tweepy
 - facebook-messenger-bot
 - python-whatsapp-bot
- Slacks, Salesforces, CRMs



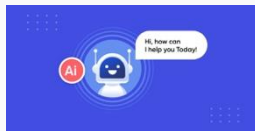
- Customer Service Systems

- Rule-based Chatbots

- Basic chatbots, such as rule-based ones, operate on predefined rules and patterns to generate responses based on specific keywords or commands.

- AI-powered Chatbots

- These, including advanced ones like ChatGPT, utilize deep learning techniques and advanced machine learning algorithms to analyze and understand natural language input, enabling them to generate contextually relevant and personalized responses.



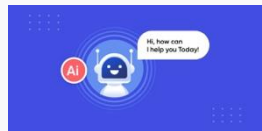
- Customer Service Systems

- Hybrid Chatbots:

- These chatbots combine elements of both rule-based and AI-powered approaches to capitalize on the strengths of each chatbot type.

- Retrieval-based Chatbots

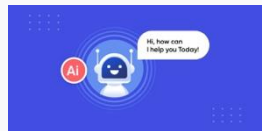
- These chatbots retrieve predefined responses from a database based on user input, making them suitable for handling structured inquiries.



- Customer Service Systems

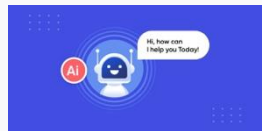
- Goals of chatbots:

- Providing quick answers to common customer queries
 - Reducing call volume to human agents by handling routine requests
 - Offering personalized recommendations and cross-sell opportunities
 - Collecting customer data and feedback to improve products and services



- Customer Service Systems

- Common applications of chatbots
 - Answering frequently asked questions to deflect support tickets
 - Providing account information such as order status or payment confirmation
 - Guiding users through processes like password resets or form - filling
 - Making recommendations based on past purchases or browsing history
 - Scheduling appointments or bookings
 - Collecting customer feedback through surveys



- Customer Service Systems

Benefits

- 24/7 Availability
- Scalability
- Efficiency
- Personalization
- Data Insights



- Customer Service Systems

- Metrics for chatbots:

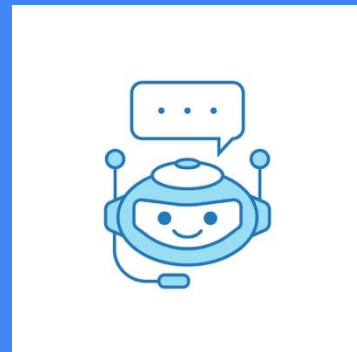
- **CSAT scores:** Customer satisfaction scores specific to chatbot interactions
 - **Containment rate:** % of inquiries resolved fully by the chatbot without human agents
 - **Conversation depth:** Avg. number of bot responses before handoff
 - **Intent recognition:** Accuracy identifying inquiry intents
 - **FAQ coverage:** % of FAQs it can address without human assistance





We Work With Data

Thank you!



ailton.moreira@openup.pt

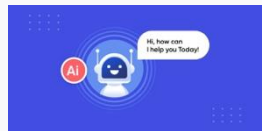


Chatbots Introduction

Ailton Moreira
ailton.moreira@openup.pt

Lisboa 2024

- Testing and Optimizations
 - How to Test Your Chatbot
 - Collecting User Feedback
 - Improving and Expanded Functionalities
- Conclusion and Next Steps
 - Future Challenges in the World of Chatbots
 - Trends and New Technologies
 - How to Stay Updated



- How to test your chatbot

- Testing chatbots effectively involves several best practices to ensure they function correctly, provide accurate responses, and deliver a good user experience.

- Unit Testing

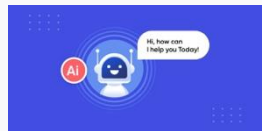
- Isolated Testing
- Mocking External Dependencies

- Integration Testing

- Testing End-to-End
- Database (truth source) and API Integration



- How to test your chatbot
 - Conversation Flow Testing
 - Simulate User Interactions
 - State Management
 - NLU Testing
 - Intent Recognition
 - Entity Extraction



- How to test your chatbot
 - Performance Testing
 - Response Time
 - Load Testing
 - User Experience Testing
 - User Satisfaction
 - Fallback and Error Handling
 - Automation and Continuous Integration
 - Automate Tests
 - Coverage Reports



- How to test your chatbot

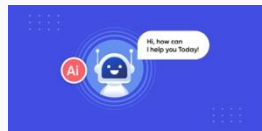
- Libraries and Tools

- **unittest**: Python's built-in library for creating and running tests.
 - **pytest**: A robust testing framework that supports fixtures, parameterized tests, and more.
 - **pytest-cov**: A plugin for pytest that measures test coverage.
 - **unittest.mock**: Part of the unittest library for mocking objects in tests.
 - **requests-mock**: For mocking HTTP requests in tests.



- Collecting user feedback

- Crucial for improving the performance, user experience and accuracy level of our chatbot.
 - Design the Feedback Mechanism
 - Implement the Feedback Collection Logic
 - Store the Feedback
 - Analyse and Evaluate the feedback
 - Take actions based on the feedback evaluation



- Improving and expanding functionalities
 - Fine-tuning ML models
 - Increase the dataset with new data
 - Retrain ML models
 - Test and train new ML models with the dataset
 - Evaluate different ML models train output
 - Integration with external APIs
 - Integration with database
 - Add more personalization capabilities



- Future challenges in the world of chatbots
 - Natural Language Understanding (NLU) and Generation (NLG)
 - Ambiguity and Context
 - Multilingual Support
 - Personalization and Context Awareness
 - Long-term Memory
 - User Preferences
 - Emotional Intelligence
 - Sentiment Analysis
 - Empathy



- Future challenges in the world of chatbots

- Ethical and Privacy Concerns
 - Data Privacy
 - Bias and Fairness
- Integration and Interoperability
 - System Integration
 - Cross-platform Consistency
- Security
 - Authentication
 - Malicious Use



- Future challenges in the world of chatbots

- Scalability and Performance
 - Handling High Volume
 - Latency
- Human-AI Collaboration
 - Hand-off Mechanisms
 - Training and Supervision
- Regulation and Compliance
 - Compliance with Laws
 - Ethical Standards



- Future challenges in the world of chatbots
 - Some strategies to address these challenges
 - Advanced NLP models
 - Human-in-the-loop systems
 - Robust data governance policies and practices
 - Cross-disciplinary collaboration
 - User feedback and iteration



- Trends and new technologies

- Trends

- Conversational AI and Natural Language Understanding (NLU)

- Advanced NLP Models
 - Contextual Understanding

- Voice-Enabled Chatbots

- Voice Assistants
 - Speech-to-Text and Text-to-Speech



- Trends and new technologies

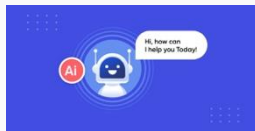
- Trends

- Multimodal Interaction

- Text, Voice, and Visuals
 - AR and VR Integration

- Personalization and Hyper-Personalization

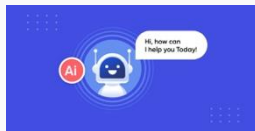
- User Profiling
 - Predictive Analytics



- Trends and new technologies
 - Trends
 - Emotional Intelligence
 - Sentiment Analysis
 - Emotion Recognition
 - Integration with Business Processes
 - CRM and ERP Systems
 - E-commerce and Customer Support
 - Security and Privacy
 - Enhanced Security Measures & Privacy Compliance



- Trends and new technologies
 - Emerging Technologies
 - Transformer-based Models
 - GPT, Gemini, and Beyond
 - Fine-Tuning and Custom Models
 - Reinforcement Learning
 - Adaptive Learning
 - Dynamic Response Generation



- Trends and new technologies
 - Federated Learning
 - Decentralized Training
 - Collaborative Improvement
 - Graph Neural Networks (GNNs)
 - Contextual Understanding
 - Knowledge Graphs
 - Explainable AI (XAI)
 - Transparency
 - User Education



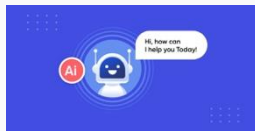
- Trends and new technologies
 - Edge Computing
 - Low Latency
 - Resource Efficiency
 - AI Ethics and Bias Mitigation
 - Fairness and Inclusivity
 - Ethical AI Practices



- How to stay updated
 - Join online communities
 - Continuous learning
 - Explore more chatbot options and new tools and technologies
 - Network with professionals



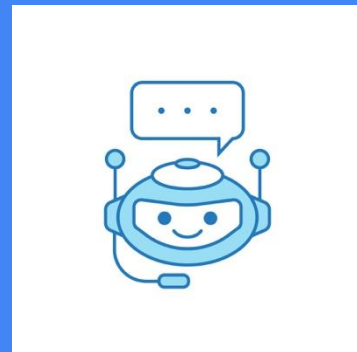
- Training evaluations
 - General training evaluation
 - Trainer evaluation
 - Evaluation of trainees
 - Appreciation of trainees





We Work With Data

Thank you!



ailton.moreira@openup.pt