Conversational AI & NLU

Understanding human-machine dialogue

ITAI 2373: Module 12

Learning Outcomes

- Define conversational AI & understand its core components and pipeline
- Master the three key NLU tasks: intent recognition, entity extraction, and state tracking
- Compare rule-based, retrieval-based, generative & RAG
- Discuss voice vs text, applications & ethics

Agenda

Conversational AI Fundamentals

NLU Deep dive

Chatbot Paradigms

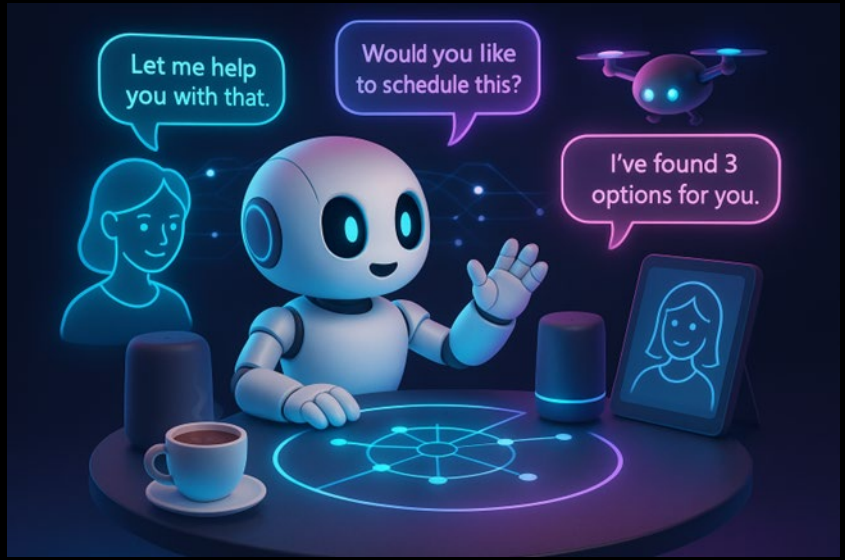
Development tool & Platforms

Interface design

Applications & Ethics

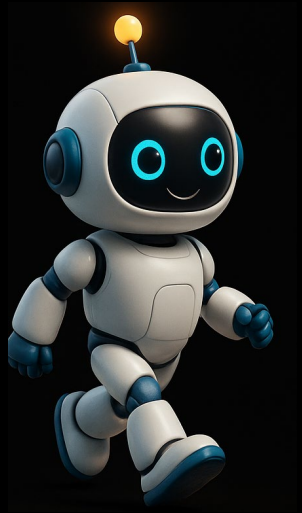
What is Conversational AI?

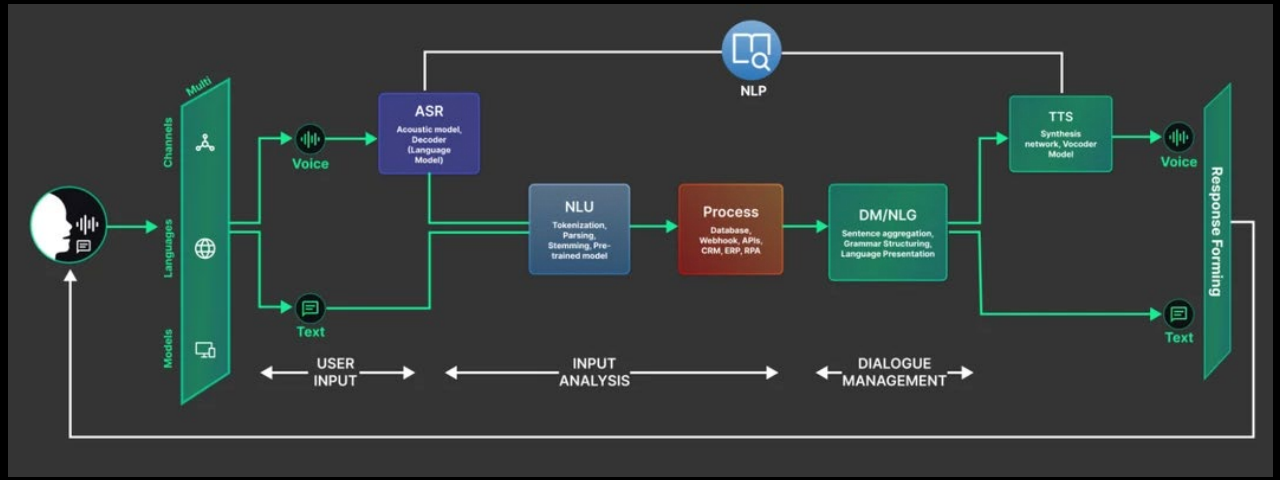
- Systems that simulate human dialogue via speech or text
- Combines speech processing, NLU, dialogue management & NLG
- Enables virtual assistants, chatbots & other interactive agents



[15]

Conversational AI Pipeline





User Input
Voice or text

Input Processing
ASR or tokenization

NLU
Intent & entity extraction

Dialogue Manager
Maintains context & plan

NLG
Generates response

Response
TTS or text output

[16]

NLU Tasks Overview:

Three core NLU tasks:

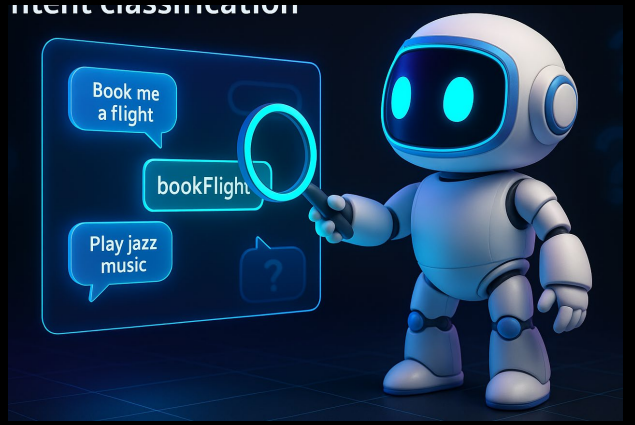
- Intent Classification,
- Entity Extraction & Slot Filling,
- Dialogue State Tracking

Multi-Turn Context:

- How these tasks work together across conversation turns

Intent Classification

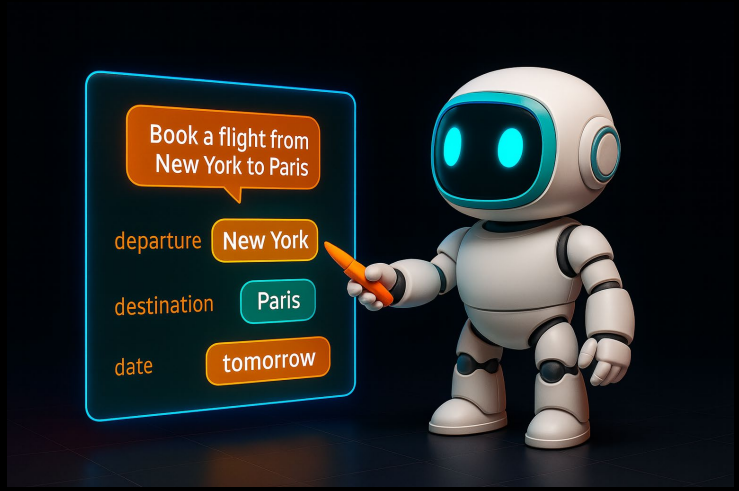
- Identify the user’s goal (e.g., bookFlight, getWeather)
- Classifiers map utterances to predefined intents
- Handle unknown or ambiguous intents via fallback



[17]

Entity Extraction & Slot Filling

- Detect entities like dates, locations, people
- Use sequence labelling to assign slot tags
- Fill slots to populate API calls or database queries



Intent: bookFlight

Book	a	flight	to	Paris	tomorrow

[18]

Dialogue State Tracking

- Maintains context across turns
- Stores intent, slot values & system actions
- Enables coherent conversation planning

[19]

Chatbot Paradigms Content:

- Four main approaches to building conversational systems

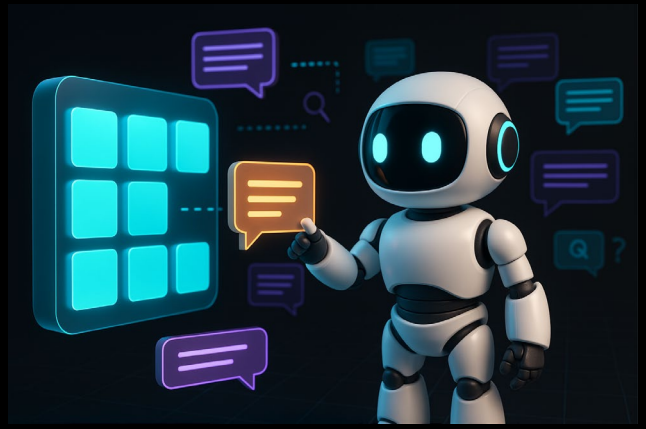
- Each with distinct strengths and use cases:
  - Rule-Based Systems
  - Retrieval-Based Systems
  - Generative Systems
  - Retrieval-Augmented Generation (RAG)

Rule-Based Chatbots

- Uses predefined rules and pattern matching
- Deterministic; predictable responses
- Limited flexibility and scalability

Retrieval-Based Chatbots

- Select responses from a database of pre-written utterances
- Use intent classification and similarity search
- Works well in closed domains (FAQs, support)



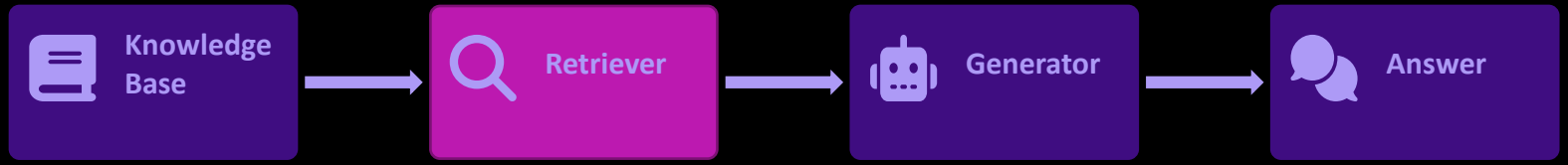
[20]

Generative Chatbots

- Use neural models to generate responses
- Flexible: can answer open-ended questions
- Risk of hallucination; less controllable

[21]

Retrieval-Augmented Generation (RAG)



RAG combines retrieval and generation

Retriever fetches relevant passages from a knowledge sourceGenerator conditions on these passages to produce grounded responsesReduces hallucination and improves factual accuracy

[22]

Choosing a Paradigm

Paradigm	Strengths	When to Use
Rule-Based	Deterministic & safe	Simple domains with predictable interactions
Retrieval	Consistent & domain-specific	FAQs and support knowledge bases
Generative	Flexible & creative	Open-ended conversations & creative tasks
RAG	Grounded & factual	Complex queries requiring external knowledge

[23] [24]

Building From Scratch – Libraries & Frameworks

Python NLP Libraries:

- **NLTK:** Natural Language Toolkit - comprehensive NLP library
- **spaCy:** Industrial-strength NLP with pre-trained models
- **Transformers (Hugging Face):** State-of-the-art pre-trained language models

Machine Learning Frameworks:

- **TensorFlow/Keras:** Google's ML framework with high-level API
- **PyTorch:** Facebook's research-friendly deep learning framework
- **scikit-learn:** Traditional ML algorithms and utilities

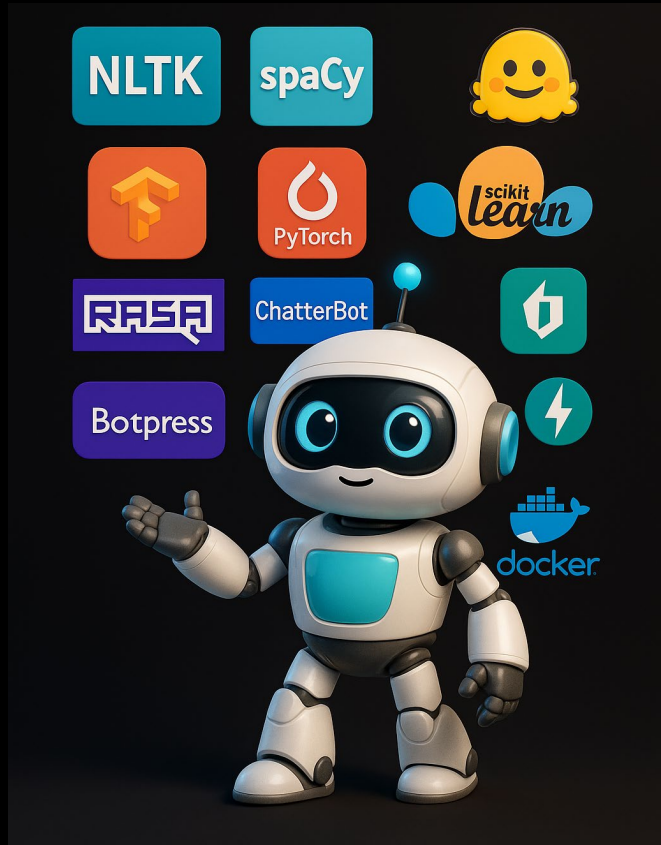
Building From Scratch – Libraries & Frameworks 2

Conversational AI Frameworks:

- **Rasa:** Open-source framework for contextual AI assistants
- **ChatterBot:** Python library for creating chatbots
- **Botpress:** Open-source conversational AI platform

Supporting Tools:

- **FastAPI/Flask:** Web frameworks for API deployment
- **SQLite/PostgreSQL:** Database options for conversation storage
- **Docker:** Containerization for deployment



Low-Code/No-Code Platforms

Microsoft Ecosystem:

- **Azure Bot Framework:** Enterprise-grade bot development platform
- **Power Virtual Agents:** No-code chatbot builder integrated with Office 365

Google Cloud:

- **Dialogflow:** Google's conversational AI platform with natural language understanding
- **Contact Center AI:** Enterprise customer service solutions

Amazon Web Services:

- **Amazon Lex:** Build conversational interfaces with automatic speech recognition
- **Amazon Connect:** Cloud-based contact center service

Low-Code/No-Code Platforms 2

NVIDIA AI:

- **Riva:** Speech AI SDK for real-time conversational AI
- **NeMo:** Toolkit for conversational AI model development

Other Notable Platforms:

- **IBM Watson Assistant:** Enterprise AI assistant platform
- **Botpress:** Open-source conversational AI platform with visual builder
- **Voiceflow:** Visual conversation design platform
  - **Speaker Notes:**

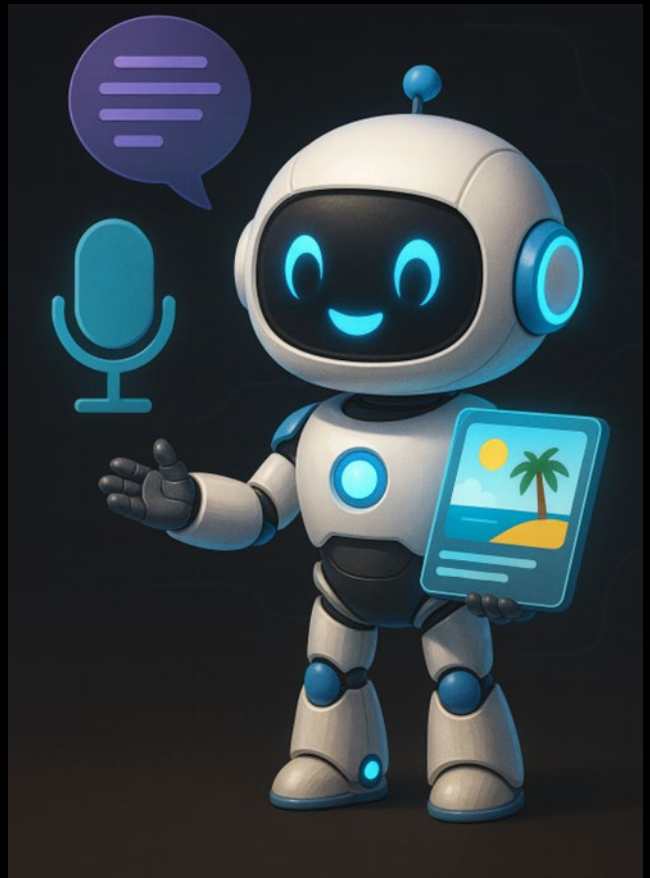
Voice VS Text Interfaces

- Voice Interface
  - Requires ASR and TTS modules
  - Handles accents, noise & speech nuances
  - Design for natural prosody & timing
- Text Interface
  - Direct text input & output
  - Parse punctuation, emojis & formatting
  - Supports multi-turn history & quick scanning

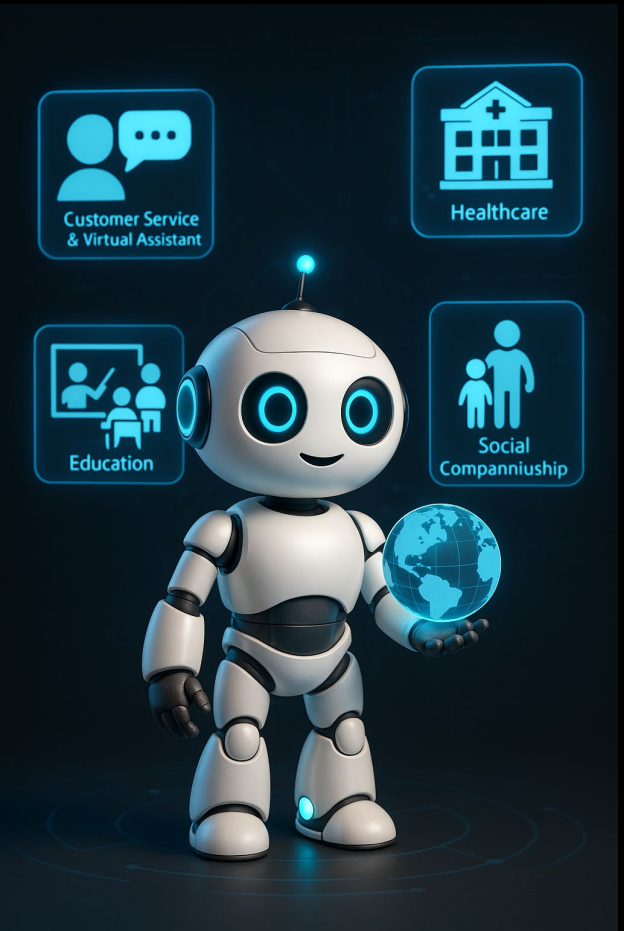
[25]

Hybrid & Multimodal Interfaces

- Support both voice and text channels
- Integrate images, gestures & other signals
- Provide richer user experiences



Real-World Impact Applications



Major Application Areas:

- Customer Service & Virtual Assistants, Healthcare, Education, Social & Companionship

Industry Impact:

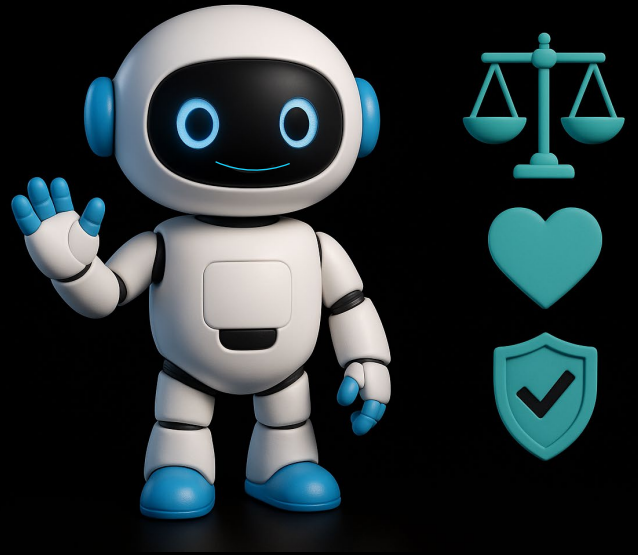
- Cost Reduction, Accessibility, Personalization, Scale

Implementation Considerations:

- Domain Specificity, Regulatory Compliance, User Expectations

Ethics & Responsible Design

- Privacy & data security
- Bias & fairness in training data
- Transparency & user consent
- Safety & human oversight



[26]

Key Takeaways

Conversational AI is an orchestration of multiple technologies working together (ASR, NLU, dialogue management, NLG, TTS)

Master the three NLU tasks: Intent detection, entity extraction, and state tracking work together like a team

Choose the right paradigm: Rule-based, retrieval-based, generative, and RAG each have optimal use cases

Development approach matters: From-scratch coding offers control, platforms offer speed - choose based on your constraints

Start with platforms, evolve to custom: Rapid prototyping with tools like Dialogflow, then customize as needed

Interface design is crucial: Voice and text require completely different design philosophies

Ethics is foundational: Privacy, fairness, transparency, and safety must be built in from day one

[27] [28] [29] [30] [31]