CSCE 580: Introduction to AI
CSCE 581: Trusted AI

Lecture 17: Building Chatbots

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

24TH OCT 2023

Student Presenters:
Kausik Lakkaraju, Bharath C Muppasani, Sara Rae Jones

Carolinian Creed: “I will practice personal and academic integrity.”

Credits: **Copyrights of all material reused acknowledged**

Organization of Lecture 17

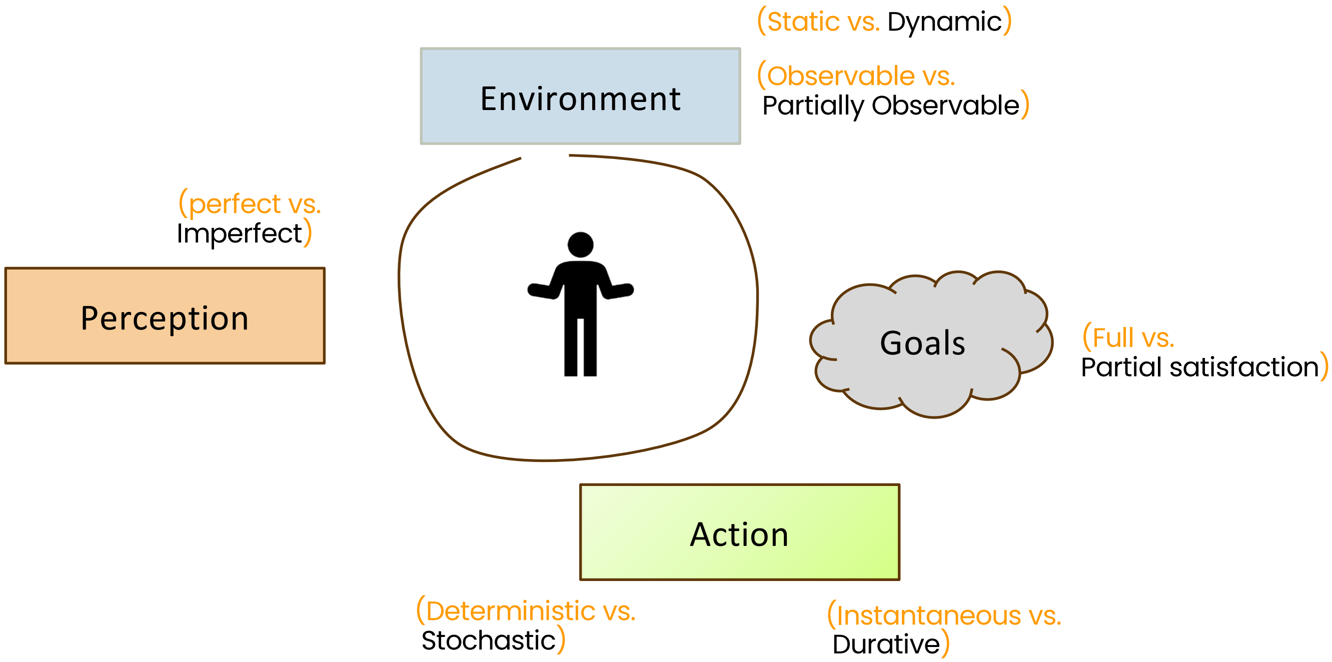
- Introduction Segment
  - Recap of Lecture 16
- Main Segment
  - Building Chatbots
    - RASA
    - SafeChat Framework
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 18
  - Ask me anything

Introduction Section

Recap of Lecture 16

- Topic discussed
  - Trust Issues
    - Explainability
  - LIME tool

Intelligent Agent Model



(Static vs. Dynamic)

Environment

(Observable vs. Partially Observable)

(perfect vs. Imperfect)

Perception

Goals

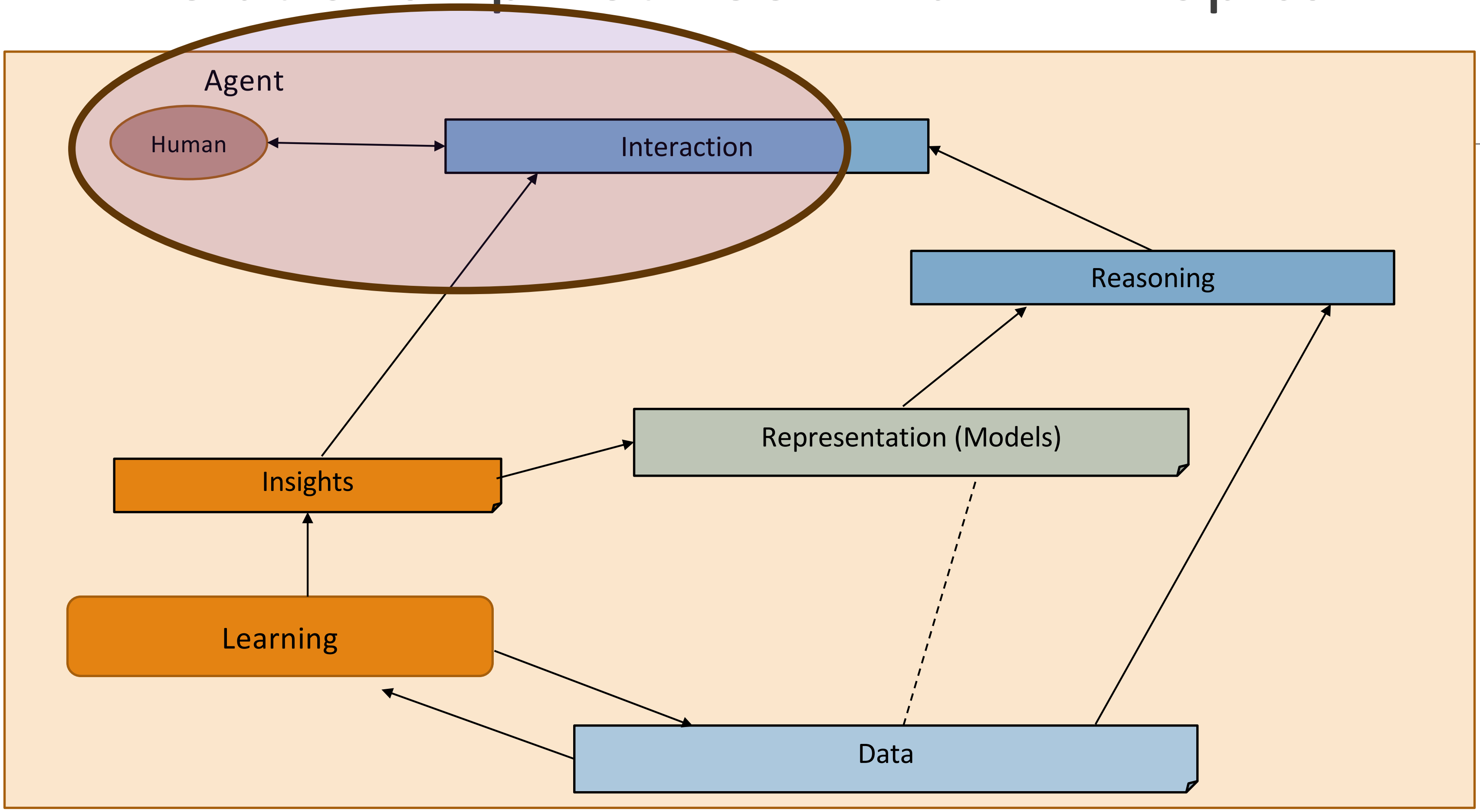
(Full vs. Partial satisfaction)

Action

(Deterministic vs. Stochastic)

(Instantaneous vs. Durative)

Relationship Between Main AI Topics



Where We Are in the Course

CSCE 580/ 581 – In This Course

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week 8: Machine Learning - Classification
- Week 9: Machine Learning - Classification – Trust Issues and Mitigation Methods
- Topic 10: Learning neural network, deep learning, Adversarial attacks
- Week 11: Large Language Models – Representation, Issues
- Topic 12: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 13: Planning, Reinforcement Learning – Sequential decision making
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Main Section

Credit: Retrieved from internet

Rasa

- Rasa is an open source, scalable AI framework that can be used to build conversational agents.
- Rasa is used by many companies like American Express, BlueCross BlueShield, Dell, …..
- Chatbot terminology:
  - **Intent**: Intention of the user behind a message.
  - **Entity**: They are used to identify important parts of the message that affects the response chosen by the chatbot. Ex: Time, location.

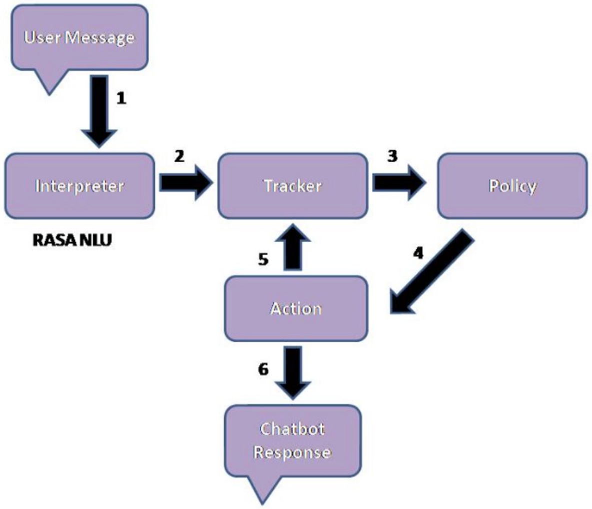
Message	Intent	Entities
When will the flight to ORD arrive at CAE?	Arrivals	ORD, CAE
Can I book an appointment with the dentist for tomorrow?	Booking	dentist, tomorrow

Credit:

1. https://rasa.com/docs/rasa/

Rasa Workflow

- Rasa consists of two major components: Rasa NLU and Rasa Core.
  - Rasa Natural Language Understanding (NLU) takes user message as input and recognizes entities and intent and passes it to RASA core.
  - Rasa Core selects the appropriate response based on the the input passed by Rasa NLU and sends it back to the user.
- Rasa workflow:
  - User message is converted into a dictionary which consists of the message, intent, and entities extracted.
  - Tracker keeps a record of the conversation.
  - Current state of the tracker is sent to the policy which decides the next action that needs to be taken.
  - Chatbot's response depends on the action chosen.



Credit:

1. https://rasa.com/docs/rasa/
2. https://www.analyticsvidhya.com/blog/2022/02/a-simple-guide-to-rasa-3-x/

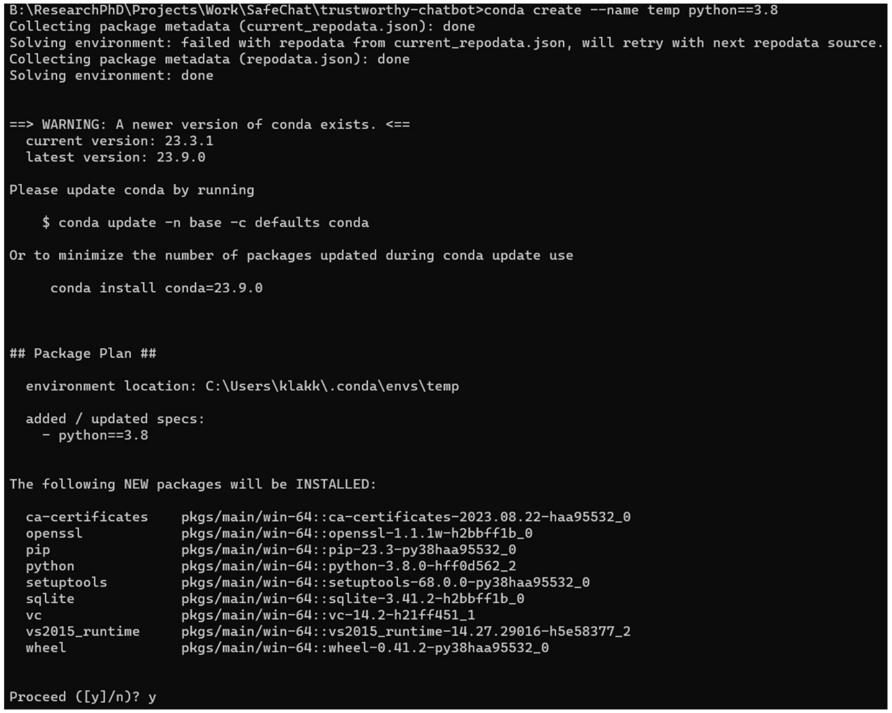
Setting Up Virtual Environment (python env)

1. If you do not have ‘pip’ on your system, download the script from: https://bootstrap.pypa.io/get-pip.py
2. Run ‘python get-pip.py’ on your terminal to install pip.
3. Run ‘python3 -m pip install --user virtualenv’ to install virtual environment package.
4. Run ‘python3 -m venv env’ to create a virtual environment, ‘env’.
5. Run ‘source env/bin/activate’ to activate the virtual environment.
6. To leave the virtual environment, run ‘deactivate’.

Alternative: use conda

Setting Up Virtual Environment (conda env; recommended)

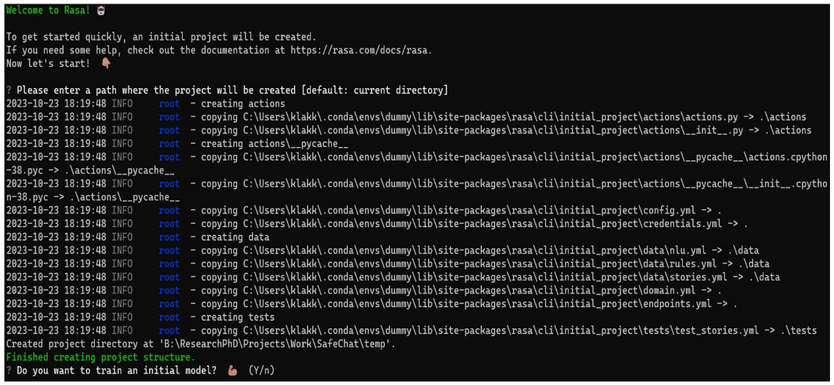
1. Install anaconda from https://www.anaconda.com/download.
2. To create a virtual environment, run the command ‘conda create –name <env_name> python==3.8’
3. To activate the environment, run ‘conda activate <env_name>’
4. To deactivate the environment, run ‘conda deactivate’.



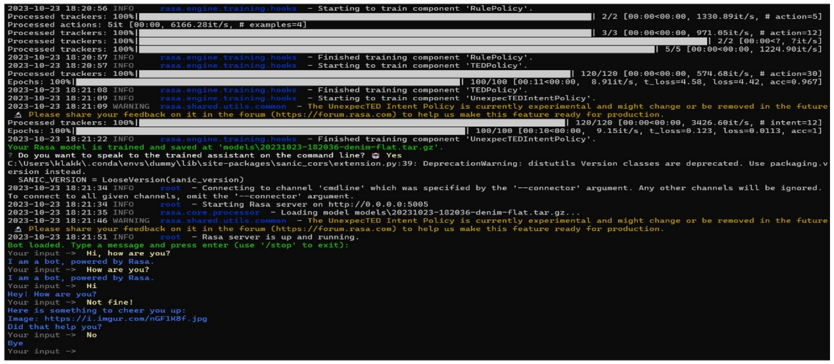
```
B:\ResearchPhD\Projects\Work\SafeChat\trustworthy-chatbot>conda create --name temp python==3.8
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.3.1
  latest version: 23.9.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=23.9.0



## Package Plan ##

  environment location: C:\Users\klakk\.conda\envs\temp

  added / updated specs:
    - python==3.8


The following NEW packages will be INSTALLED:

  ca-certificates    pkgs/main/win-64::ca-certificates-2023.08.22-haa95532_0
  openssl            pkgs/main/win-64::openssl-1.1.1w-h2bbff1b_0
  pip                pkgs/main/win-64::pip-23.3-py38haa95532_0
  python             pkgs/main/win-64::python-3.8.0-hff0d562_2
  setuptools         pkgs/main/win-64::setuptools-68.0.0-py38haa95532_0
  sqlite             pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
  vc                 pkgs/main/win-64::vc-14.2-h21ff451_1
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
  wheel              pkgs/main/win-64::wheel-0.41.2-py38haa95532_0


Proceed ([y]/n)? y
```

Build a Chatbot with Rasa

1. Install Rasa using the command, 'pip install rasa==3.1'
2. Run 'rasa init' to build a basic Rasa chatbot.
3. Run 'rasa shell' to talk to your chatbot.



```
Welcome to Rasa! 🤖

To get started quickly, an initial project will be created.
If you need some help, check out the documentation at https://rasa.com/docs/rasa.
Now let's start! 👇

? Please enter a path where the project will be created [default: current directory]
2023-10-23 18:19:48 INFO     root  - creating actions
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\actions\actions.py -> .\actions
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\actions\__init__.py -> .\actions
2023-10-23 18:19:48 INFO     root  - creating actions\__pycache__
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\actions\__pycache__\actions.cpython
-38.pyc -> .\actions\__pycache__
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\actions\__pycache__\__init__.cpytho
n-38.pyc -> .\actions\__pycache__
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\config.yml -> .
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\credentials.yml -> .
2023-10-23 18:19:48 INFO     root  - creating data
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\data\nlu.yml -> .\data
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\data\rules.yml -> .\data
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\data\stories.yml -> .\data
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\domain.yml -> .
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\endpoints.yml -> .
2023-10-23 18:19:48 INFO     root  - creating tests
2023-10-23 18:19:48 INFO     root  - copying C:\Users\klakk\.conda\envs\dummy\lib\site-packages\rasa\cli\initial_project\tests\test_stories.yml -> .\tests
Created project directory at 'B:\ResearchPhD\Projects\Work\SafeChat\temp'.
Finished creating project structure.
? Do you want to train an initial model? 💪  (Y/n)
```



```
2023-10-23 18:20:56 INFO     rasa.engine.training.hooks  - Starting to train component 'RulePolicy'.
Processed trackers: 100%|██████████████████████████████| 2/2 [00:00<00:00, 1330.89it/s, # action=5]
Processed actions: 5it [00:00, 6166.28it/s, # examples=4]
Processed trackers: 100%|██████████████████████████████| 3/3 [00:00<00:00, 971.05it/s, # action=12]
Processed trackers: 100%|██████████████████████████████████████| 2/2 [00:00<?, ?it/s]
Processed trackers: 100%|██████████████████████████████████| 5/5 [00:00<00:00, 1224.90it/s]
2023-10-23 18:20:57 INFO     rasa.engine.training.hooks  - Finished training component 'RulePolicy'.
2023-10-23 18:20:57 INFO     rasa.engine.training.hooks  - Starting to train component 'TEDPolicy'.
Processed trackers: 100%|█████████████████████████████| 120/120 [00:00<00:00, 574.68it/s, # action=30]
Epochs: 100%|██████████████████████████| 100/100 [00:11<00:00,  8.91it/s, t_loss=4.58, loss=4.42, acc=0.967]
2023-10-23 18:21:08 INFO     rasa.engine.training.hooks  - Finished training component 'TEDPolicy'.
2023-10-23 18:21:09 INFO     rasa.engine.training.hooks  - Starting to train component 'UnexpecTEDIntentPolicy'.
2023-10-23 18:21:09 WARNING  rasa.shared.utils.common  - The UnexpecTED Intent Policy is currently experimental and might change or be removed in the future
 ⚠ Please share your feedback on it in the forum (https://forum.rasa.com) to help us make this feature ready for production.
Processed trackers: 100%|█████████████████████████████| 120/120 [00:00<00:00, 3426.60it/s, # intent=12]
Epochs: 100%|██████████████████████████| 100/100 [00:10<00:00,  9.15it/s, t_loss=0.123, loss=0.0113, acc=1]
2023-10-23 18:21:22 INFO     rasa.engine.training.hooks  - Finished training component 'UnexpecTEDIntentPolicy'.
Your Rasa model is trained and saved at 'models\20231023-182036-denim-flat.tar.gz'.
? Do you want to speak to the trained assistant on the command line? 🤖 Yes
C:\Users\klakk\.conda\envs\dummy\lib\site-packages\sanic_cors\extension.py:39: DeprecationWarning: distutils Version classes are deprecated. Use packaging.v
ersion instead.
  SANIC_VERSION = LooseVersion(sanic_version)
2023-10-23 18:21:34 INFO     root  - Connecting to channel 'cmdline' which was specified by the '--connector' argument. Any other channels will be ignored.
To connect to all given channels, omit the '--connector' argument.
2023-10-23 18:21:34 INFO     root  - Starting Rasa server on http://0.0.0.0:5005
2023-10-23 18:21:35 INFO     rasa.core.processor  - Loading model models\20231023-182036-denim-flat.tar.gz...
2023-10-23 18:21:46 WARNING  rasa.shared.utils.common  - The UnexpecTED Intent Policy is currently experimental and might change or be removed in the future
 ⚠ Please share your feedback on it in the forum (https://forum.rasa.com) to help us make this feature ready for production.
2023-10-23 18:21:51 INFO     root  - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  Hi, how are you?
I am a bot, powered by Rasa.
Your input ->  How are you?
I am a bot, powered by Rasa.
Your input ->  Hi
Hey! How are you?
Your input ->  Not fine!
Here is something to cheer you up:
Image: https://i.imgur.com/nGF1K8f.jpg
Did that help you?
Your input ->  No
Bye
Your input ->
```

What is SafeChat?

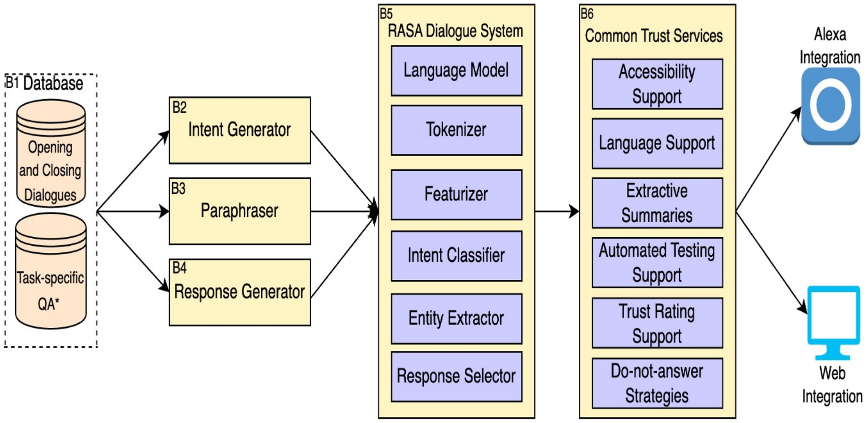
- SafeChat architecture was introduced in [1]. It is a Rasa-based framework that can be used to build safe and trustworthy chatbots.
- The uniqueness of SafeChat is:
  - A safe design where the responses can be traced back to their original source (e.g., official FAQs).
  - A do-not-answer strategy that can deflect certain user questions that are not supposed to be answered.
  - A low-programming design pattern based on the open-source Rasa platform to generate chatbots quickly for any region.
  - A quick domain-independent chatbot framework with CSV-based Q/A support and automatic intent generator with support for backend integration and testing.

References:

1. Muppasani, B., Pallagani, V., Lakkaraju, K., Lei, S., Srivastava, B., Robertson, B., … & Narayanan, V. (2022). On Safe and Usable Chatbots for Promoting Voter Participation. *arXiv preprint arXiv:2212.11219*.

SafeChat Architecture (1/2)

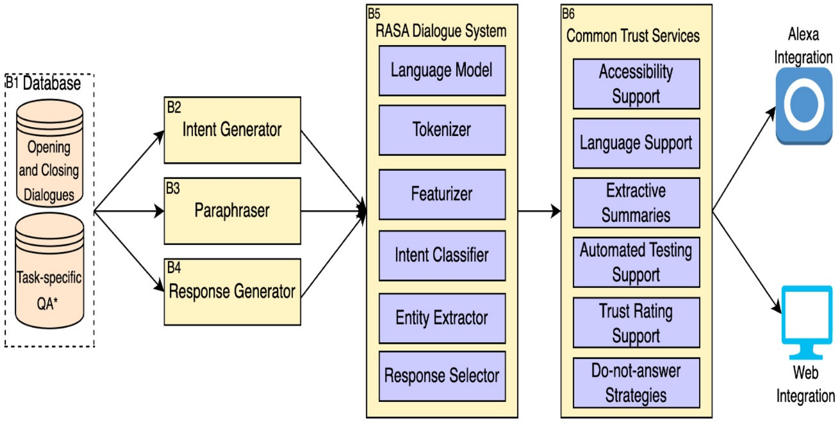
- **Database (B1):** The database is the source from which we extract the training data to train the chatbot. We ensure that the source is reliable and trustworthy. Task-specific QA refers to the data source pertaining to the chosen domain. The opening and closing dialogues are usually generic (like greeting and saying bye).
- **Intent Generator (B2):** Intent Generator helps in tagging existing questions to an intent, which can later be utilized to map any new incoming user utterance to an available intent to provide desired answers.
- **Paraphraser (B3):** A paraphraser can be used to augment the training data by paraphrasing the questions given in an official FAQ document.



*__Domains__: Elections, Finance, Water, Education, Power, ...

SafeChat Architecture (2/2)

- **Response Generator (B4):** A response is usually text but can also include multi-modal content like images and audio. The safe chatbot architecture reuses the response generation module available in the RASA Dialogue System.
- **RASA Dialogue System (B5):** We use the RASA chatbot framework to build the chatbot. The dialogue system has an NLU pipeline with different components for understanding human conversation and responding appropriately.
- **Common Services (B6):** The common services are optional, and the user has the flexibility of choosing the services they need. Some of the accessibility options are font settings and Text-to-Speech.
- **System Integration:** Our framework allows easier web integration and integration with Alexa.



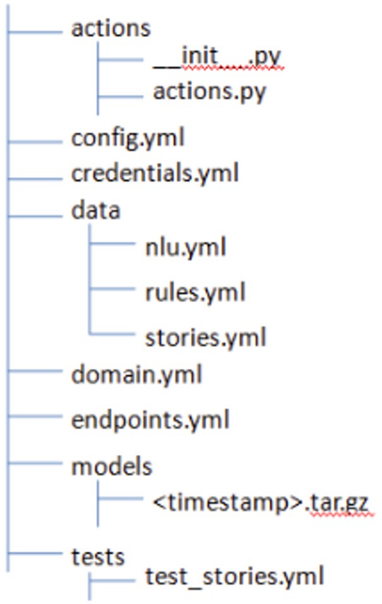
***Domains**: Elections, Finance, Water, Education, Power, ...

Building a Chatbot using SafeChat

1. Run ‘git clone https://github.com/ai4society/trustworthy-chatbot.git’ to clone our SafeChat repository.
2. Go to the project directory and run ‘pip install –r requirements.txt’ to install all the required packages.
3. Run ‘code/configure_rasa.py’, ’code/extract_intent.py’ and ‘code/paraphraser.py’ files in the same order to create your chatbot.

Chatbot File Structure

- config.yml: It has pipeline, policies, and other NLU components.
- '/data' consists of:
  - nlu.yml: It contains intents and examples for each intent with entities.
  - rules.yml: Defines a fixed conversation path that chatbot follows.
  - stories.yml: It contains conversations paths the chatbot needs to take in order to respond to the user messages appropriately.
- domain.yml: This file acts as an index for the chatbot. It contains intents, entities, slots, responses, forms, and actions.
- '/models' contain the trained chatbot model that can be used to talk to the chatbot.



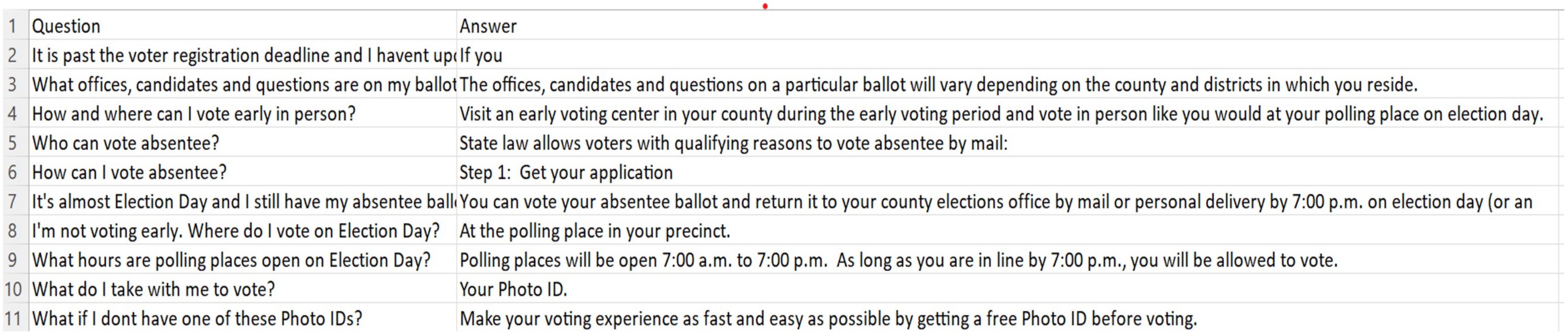
```
├── actions
│     ├── __init__.py
│     └── actions.py
├── config.yml
├── credentials.yml
├── data
│     ├── nlu.yml
│     ├── rules.yml
│     └── stories.yml
├── domain.yml
├── endpoints.yml
├── models
│     └── <timestamp>.tar.gz
└── tests
      └── test_stories.yml
```

Credit:
1. https://rasa.com/docs/rasa/
2. https://www.analyticsvidhya.com/blog/2022/02/a-simple-guide-to-rasa-3-x/

Running your chatbot

1. Go to the ‘Chatbot’ directory that is generated and to train the chatbot, run ‘rasa train’.
2. To talk to the trained chatbot, run ‘rasa shell’.

Build your own chatbot with SafeChat



1	Question	Answer
2	It is past the voter registration deadline and I havent up	If you
3	What offices, candidates and questions are on my ballot	The offices, candidates and questions on a particular ballot will vary depending on the county and districts in which you reside.
4	How and where can I vote early in person?	Visit an early voting center in your county during the early voting period and vote in person like you would at your polling place on election day.
5	Who can vote absentee?	State law allows voters with qualifying reasons to vote absentee by mail:
6	How can I vote absentee?	Step 1: Get your application
7	It's almost Election Day and I still have my absentee ball	You can vote your absentee ballot and return it to your county elections office by mail or personal delivery by 7:00 p.m. on election day (or an
8	I'm not voting early. Where do I vote on Election Day?	At the polling place in your precinct.
9	What hours are polling places open on Election Day?	Polling places will be open 7:00 a.m. to 7:00 p.m. As long as you are in line by 7:00 p.m., you will be allowed to vote.
10	What do I take with me to vote?	Your Photo ID.
11	What if I dont have one of these Photo IDs?	Make your voting experience as fast and easy as possible by getting a free Photo ID before voting.

1. Create a new ‘Chat.csv’ file in the ‘data/’ directory with your desired FAQs. The CSV should have a column called ‘Question’ with all the queries and another column called ‘Answer’ with the corresponding answers.
2. Run ‘code/configure_rasa.py’, ’code/extract_intent.py’ and ‘code/paraphraser.py’ files in the same order to create your chatbot.

Survey Link

Please fill the survey using the following link to help us improve our SafeChat architecture.

Course Project

Project Discussion: What Problem Fascinates You ?

- Data
  - Water
  - Finance
  - …
- Analytics
  - Search, Optimization, Learning, Planning, …
- Application
  - Building chatbot
- Users
  - Diverse demographics
  - Diverse abilities
  - Multiple human languages

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

Project Discussion: Dates and Deliverables

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
- Nov 30, 2023
  - Project report due
- Dec 5 / 7, 2023
  - In-class presentation

Skeleton: A Basic Chatbot

- Run in an infinite loop until the user wants to quit
- Handle any user response
  - User can quit by typing "Quit" or "quit" or just "q"
  - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – "I do not know this information".
- Handle known user query types // Depends on your project
  - "Tell me about N-queens", "What is N ?"
  - "Solve for N=4?"
  - "Why is this a solution? "
- Handle chitchat // Support at least 5, extensible from a file
  - "Hi" => **"Hello"**
  - …
- ***Store session details in a file***

Illustrative Project

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Discussion: Illustration

1. Create a private Github repository called "CSCE58x-Fall2023-<studentname>-Repo". Share with Instructor (biplav-s) and TA (kausik-l)
2. Create Google folder called "CSCE58x-Fall2023-<studentname>-SharedInfo". Share with Instructor (prof.biplav@gmail.com) and TA (lakkarajukausik90@gmail.com)
3. Create a Google doc in your Google repo called "Project Plan" and have the following by next class (Sep 5, 2023)

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Illustration: N-Queens

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
    - Method 1: Random solution
    - Method 2: Search – BFS
    - Method 3: Search - …
  - Human interaction: Develop a basic chatbot (no AI) as outlined
  - Deliverable
    - Code structure in Github
      - ./data
      - ./code
      - ./docs
      - ./test
    - Presentation: Make sprint presentation on Oct 12, 2023

Reference: Project Rubric

- **Project results** – 60%
  - Working system ? – 30%
  - Evaluation with results superior to baseline? – 20%
  - Considered related work? – 10%
- **Project effort**s – 40%
  - Project report – 20%
  - Project presentation (updates, final) – 20%

- **Bonus**
  - Challenge level of problem – 10%
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

Lecture 17: Summary

- We talked about
  - Building Chatbots
  - Rasa
  - SafeChat Framework

Concluding Section

About Next Lecture – Lecture 18

Lecture 18: Explanation, Machine Learning – Unsupervised

- Recap: Trusted AI/ Explanations
- Unsupervised ML/ Clustering