QA FQA Bot: Streamlining Answers with AQAC

Natural Language Processing

Beesan Alattal 0214632

Besan Musallam 0213890

Layan Balbisi 0215423

Zaina Abunasser 0218080

Leen Samman 0219463

Spring Semester 2023/2024

Supervised by Prof. Mohammad A. M. Abushariah

Computer Information System department

King Abdullah || School of Information Technology
University of Jordan

Contents

1.0 Introduction

1.1 Need Analysis and Description

The need of conversational agents has become acute with the widespread use of personal machines with the wish to communicate and the desire of their makers to provide natural language interfaces

[3].

As technology becomes increasingly important in our daily lives, businesses are relying more on digital services to assist their customers. This means there is a greater demand for computer systems that can interact with customers in a way that is both easy to understand and tailored to their needs. Customers are looking for clear and personalized responses, and businesses need to keep up with this expectation, therefor they are embracing technology to enhance communication with customers, ultimately making interactions simpler and more personalized. That's where artificial intelligence comes into play. AI-powered systems, such as chatbots, have the ability to converse with customers just like humans do, resulting in smoother and more efficient interactions. According to the English lexicon, a chatbot is defined as" a computer program designed to respond with conversational or informational replies to verbal or written messages from users" [5]. Furthermore, it's worth noting that any chatbot program is proficient in understanding one or more human languages through Natural Language Processing (NLP) [9].

In this paper, we present our work on developing a chatbot system for the University of Jordan (JU) Quality Assurance (QA) department. This chatbot is designed to assist university employees and students by addressing their questions regarding report templates, instructions, and various other services.

The chatbot was built using RASA, an open-source framework known for its natural language understanding (NLU) and Core tools that facilitate the creation of conversational systems [4]. We chose RASA because it allows for easy integration with other systems, thanks to its modular architecture [4]. This integration capability and modular design make RASA an ideal choice for building efficient and scalable chatbot solutions.

1.2 Project Constraints

We were faced several constrain while developing the chatbot, two of them is mentioned below:

- Limited Data Availability: The most important constraint was that there wasn't an available data to use. Therefore, the dataset was made from scratch, for the specific questions and conversations we expected within the QA department, which required additional time and resources.
- Technical Challenges: its correct that RASA offers strong natural language understanding capabilities, which is helpful to build reliable chatbot, but its restricted code format created challenges during the coding phase. this technical constrain required a lot of debugging efforts.

1.3 System Environment

A trustworthy operational system environment was necessary for the growth and release of chatbot for the Quality Assurance Department of University of Jordan. By integrating a self-made webpage with the chatbot using the RASA open-source framework, accessibility and user-friendliness were assured. For The training of this chatbot, a custom dataset was employed, mostly in Arabic language. The complete deployment is however preceded by an extensive testing using RASA testing tool to discover potential bugs.

1.4 Project Software and Hardware Requirements

The following Hardware and software requirements were needed to develop and deploy the chatbot: Hardware: a laptop with intel core i7 and 16 GB RAM , to get fast performance. Software: Rasa open-source framework to build the chatbot's model , and HTML to develop the webpage interface.

2.0 Research Background and Related Works

- Web-based chatbot for Frequently Asked Queries (FAQ) in Hospitals, Mamta Mittal et al. present a comprehensive web-based application designed to facilitate user interaction with a chatbot through both voice and text inputs. This application integrates text-to-speech (TTS) and speech-to-text (STT) conversion functionalities, enabling seamless communication. The chatbot leverages natural language processing (NLP) techniques such as stemming, tokenization, and

enumeration to interpret human language. It utilizes a bag-of-words approach to identify grammatically incorrect or incomplete sentences. User inputs are matched with predefined patterns of commonly asked queries, and corresponding responses are retrieved from a database. If a direct match is not found, machine learning algorithms are employed to generate suitable responses based on similar patterns or historical data. Gradient descent is used for optimization and training, enabling the chatbot to improve its performance over time through user interactions and feedback. [12]

- College FAQ Chatbot, the ”Dexter the College FAQ Chatbot” paper focuses on developing a chatbot aimed at addressing common college-related queries. This chatbot provides a user-friendly solution for students to access pertinent information round-the-clock, alleviating the need to approach faculty or staff directly. The chatbot is built using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, enabling it to formulate responses even when specific answers are not available in the database. Preliminary results indicate promising capabilities, with further training expected to enhance the chatbot’s accuracy and effectiveness in handling a broader range of questions. [7]

- Building a Chatbot on a Closed Domain using RASA(2020), this paper discusses the development of a retrieval-based chatbot on a closed domain with short conversations, which can answer questions even if they contain incorrect syntax or misspellings, but only from the training data. The chatbot was built for the

Vietnamese language and specifically designed for students at CICT of Can Tho University. The RASA framework was used to build the chatbot, and several algorithms such as SVM, CRF, KNN, and LSTM were utilized to classify intents, extract entities, predict correct entities, and manage dialogue, respectively. The chatbot architecture consists of three components: Natural Language Understanding (NLU), Dialog Management (DM), and Message Generator (MG). The CRF model showed good results but only for input containing correct entities. To improve the model's performance, new words that are misspelled were generated to help the model extract false entities. KNN was then used to convert the incorrect entities into correct entities. The model was tested using different kernels, and the 'rbf' kernel achieved the best accuracy of .33% .94 The CRF model achieved an accuracy of 95%. [10]

- University FAQ Chatbot, Bhavika R. Ranoliya, Nidhi Raghuwanshi, and Sanjay Singh from Manipal University proposed an interactive chatbot to provide students with academic information at Manipal University. This chatbot addresses queries related to university ranking, service availability, campus environment updates, and campus activities. It utilizes Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) to answer FAQs interactively. The chatbot can be enhanced by updating patterns and templates for general queries, providing more accurate responses to both general and service-based questions, thereby ensuring user satisfaction. AIML, based on XML, structures the chatbot's behavior through categories and tags, which include

patterns for input and templates for responses. AIML encompasses words, underscores, symbols, and wildcard symbols, along with optional contexts like <that> and <topic>, categorized into Atomic, Default, and Recursive categories. Future work could involve blending AIML and LSA for more natural client interactions. [10]

- In 2020, a paper titled “An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU” was published by Anran Jiao from Nankai University. The paper takes you through the process of implementing a chatbot system customized to answer finance-related questions via utilizing RASA NLU and Using Neural Networks. The chatbot starts operating from the moment the user types a question. RASA NLU extracts entities from the entered question (message), enabling the bot to spot user intents and provide appropriate responses, note that the chatbot is connected to WeChat using iex-finance API. The paper has compared two methods for the system (RASA NLU and RNN) in recognition accuracy and integrities of entity or sentence, and in a single experiment the results showed that RASA NLU method has higher accuracy compared with NN, considering one sentence as a whole, RASA NLU method is superior to extract all the entities, However, the NN method has better integrity to classify entities from segmented words. [8]

3.0 Proposed Methodology

This section outlines the proposed methodology for developing the chatbot system for the University of Jordan (JU) Quality Assurance (QA) department. The methodology includes the overall pipeline, technical implementation details, dataset description, data preprocessing, and feature classification.

3.1 Pipeline of the Proposed Methodology

The pipeline of the proposed methodology consists of several key stages:

Figure 1: Methodology Pipeline

- Data Collection To train our RASA model, we gathered information from the Jordan University Quality Assurance website. A detailed description of the dataset will be provided in later section. Data Preprocessing Cleaning and preparing our data for analysis by making sure there are no duplicates on the intents and examples.

- Model Training Using the selected features to train the chatbot model.

- Evaluation and Testing In this phase we applied unit testing to make sure the indentation is correct to assess the performance of the trained model.

- Deployment In the deployment phase, we created a user-friendly web interface for our RASA chatbot using HTML, CSS, and JavaScript. This interface features a responsive design with a navigation bar that includes the University of Jordan Quality Assurance logo and a theme toggle switch. The central chatbot container includes a chat window for conversation history and an input bar for user messages, supporting both button clicks and the Enter key for quick message sending. User and bot messages are clearly differentiated to ensure an intuitive and engaging user experience, making the chatbot easily accessible and effective in facilitating communication with the JU QA department.2 .3 Technical and Implementation Description.

3.2 Technical and Implementation Description

As mentioned before, we used the RASA framework to develop our conversational agent, making it easier for users to get the information they need through a prompt rather than searching for attributes one by one. RASA provides coherent responses based on the ongoing dialogue through two primary components: Rasa NLU and Rasa Core. Rasa NLU can be just treated like an ear which is taking inputs from user and Rasa Core is just like the brain which will take decisions based on user input[]. The approach used in our project is When the end user sends a message, it is first processed by the Chatbot, which forwards it to the Rasa NLU for intent recognition and entity

extraction. The Rasa NLU generates a structured output containing the original text, the identified intent, and any extracted entities. This structured output is then passed to the Tracker, which maintains the conversation state. The Tracker forwards this information to the Policy, which evaluates the current state of the conversation. Based on this evaluation, the Policy determines the appropriate next action to take. The selected action is logged by the Tracker. Finally, the Tracker generates and delivers the appropriate response to the user, completing the interaction.

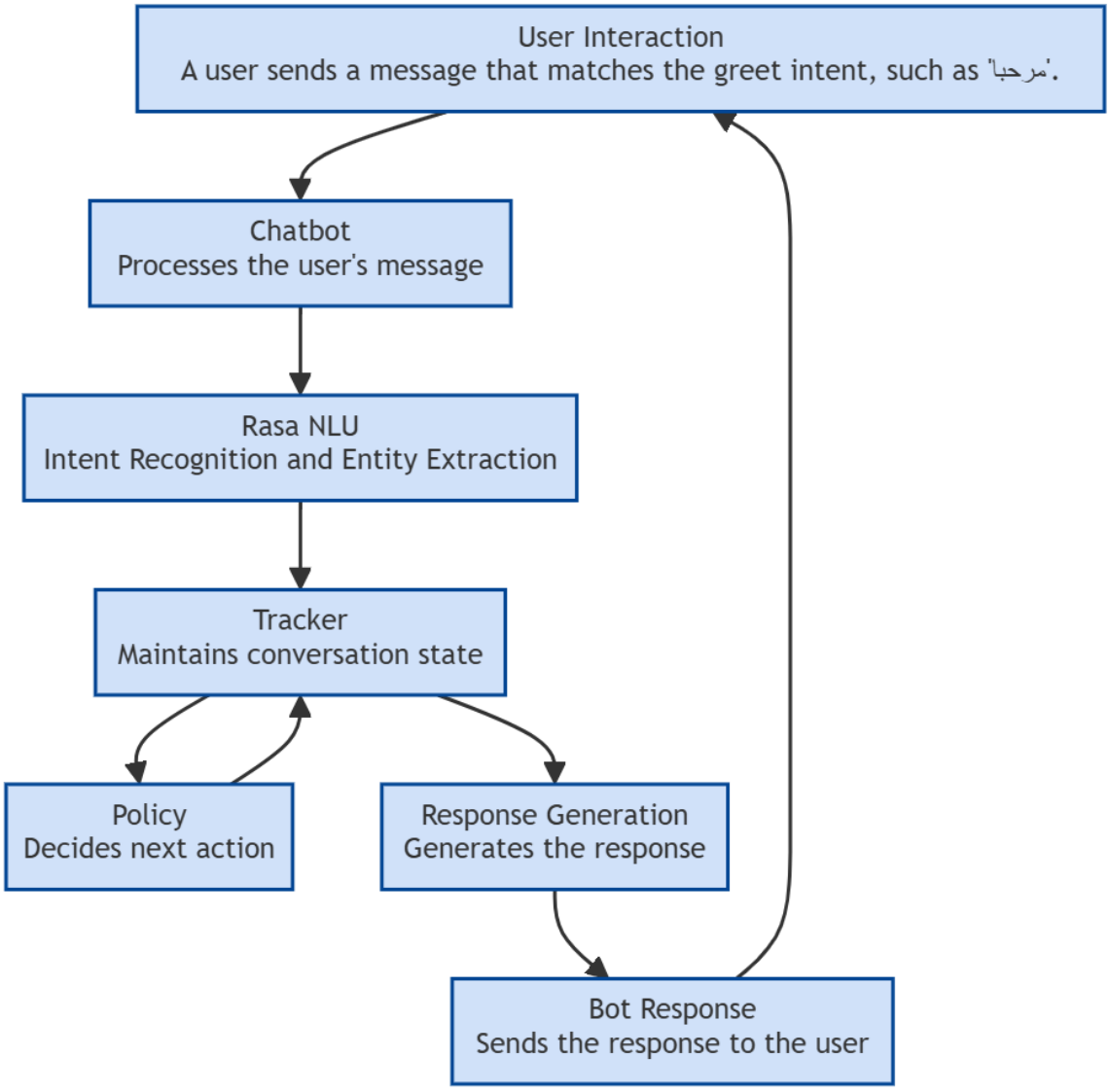

Figure 2: How RASA operates

RASA understands and correctly classifies user inputs, enabling the chatbot to respond appropriately by defining intents and providing examples in nlu.yml.

Let us break down some important basics to understand the process. here is an example of the intent

```
nlu:
- intent: greet
  examples: |
    - مرحبا
    - صباح الخير
    - كيف حالك
    - السلام عليكم
    - مساء الخير
```

Figure 3:An example of an Intent

greet that captures user inputs that are typically greetings. These are examples “sample phrases” that users might say to greet the chatbot. They help the model learn what constitutes a greeting. Here are the examples provided:

"مرحبا" (Hello)

"صباح الخير" (Good morning)

"كيف حالك" (How are you?)

"السلام عليكم" (Peace be upon you)

"مساء الخير" (Good evening)

The examples provided serve as training data for the RASA NLU model. By learning from these examples, the model becomes better at recognizing variations in how users might express the same intent. This allows the model to classify user inputs accurately and respond appropriately. When a user sends a message to the chatbot, RASA NLU parses the message and tries to match it to one of the defined intents based on the examples provided. If the message matches the examples under greet, it will classify the message as the greet intent. Based on the identified intent, RASA Core then decides on the appropriate response or action to take, as defined in the stories.yml and domain.yml files.

In stories.yml, we defined a story named" happy path," which is a simple example of a conversation flow. The steps section lists the sequence of actions in the conversation. The intent greet represents a user sending a message that matches the greet intent. The action utter_greet represents the bot's response to the greet intent, which is to perform the action utter_greet. The domain.yml file defines.

```yaml
stories:

- story: happy path
  steps:
  - intent: greet
  - action: utter_greet
```

Figure 4: An example of a Story

the chatbot's domain, including intents, entities, slots, responses, and actions, acting as a blueprint for what the bot can understand and how it should respond. For example,

the intents section lists the intents that the chatbot can recognize, such as greet, which corresponds to user messages that are greetings. The responses section defines the responses the bot can give, each with a unique identifier referenced in the stories. The utter_greet response is triggered when the bot needs to greet the user, and the text : "أهلاً بك , كيف يمكنني مساعدتك؟" is the message the bot sends when responding with utter_greet.

```
intents:
  - greet
responses:
  utter_greet:
  - text: "أهلاً بك , كيف يمكنني مساعدتك؟"
```

Figure 5:An example of an Utter

How files manage to work together:

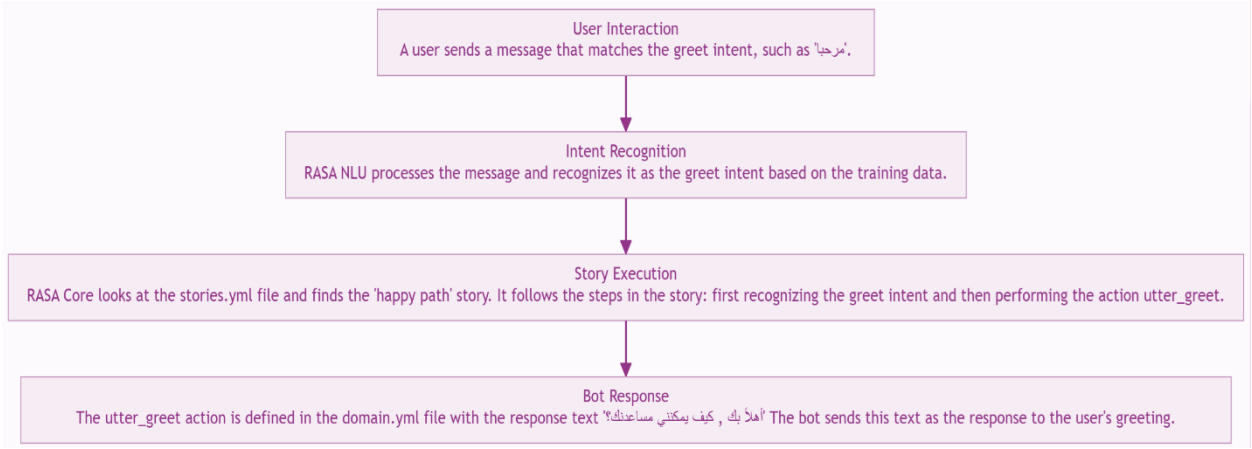

Figure 6: Caption

3.3 Dataset Description

To train our RASA model, we gathered our information independently using the Jordan University Quality Assurance website. We identified the information that users are

most likely to seek on the website or would need to call the Quality Assurance department to obtain. The data we collected includes the main topics that users might need assistance with, and potential ways users might phrase their queries when speaking to a real person. This ensures that our chatbot can effectively understand and respond to user inquiries by simulating real-world interactions and addressing the most relevant and frequently asked questions. Additionally, we collected sample queries by simulating potential user questions based on the information available on the website. For example, for the intent InformationFormForParticipantsInMandatoryCourses, we identified various ways users might ask about information forms for participants in mandatory courses, such as "نمذج المعلومات للمشاركين بالدورات الإلزامية" and " أسماء المشاركين بالدورات الالزامية".

3.4 Data Preprocessing

Data preprocessing was a crucial step in training our RASA model to ensure that the data was clean, relevant, and structured appropriately for effective training.. We ensured there were no duplicates in the intents and examples. Data cleaning involved removing irrelevant or noisy data, correcting typographical errors, and standardizing formats. We also performed tokenization, which involved breaking down text into smaller tokens, and lemmatization, which reduced words to their base or root form to improve generalization. These preprocessing steps helped in improving the performance and accuracy of the chatbot model.

By following these preprocessing steps, we ensured that our training data was of high quality and well-structured, allowing the RASA model to learn effectively and perform accurately in real-world scenarios.

3.5 Methods used in evaluation

RASA provides some metrics by using the command RASA test. The clarification of these metrics used in evaluation is provided below Accuracy = (TP + TN) / (TP + FP + TN + FN) The accuracy metric measures the ratio of correct predictions over the total number of instances evaluated Precision = (TP) / (TP + FP) Precision is the ratio of correctly predicted positive observations to the total predicted positives. F1-score=2((precision *recall)(precision+recall)) F1-Score is the weighted average of Precision and Recall. Since Recall is not directly provided in metrics RASA test provide here is the equation for the recall Recall = TP / (TP + TN).

3.6 Features Classification

Feature classification is a pivotal process in our RASA model, where the extracted features are used to classify user intents and recognize entities within the text. Intent classification determines the purpose behind a user's message using built-in algorithms in RASA, such as the Dual Intent and

Entity Transformer (DIET) classifier, which simultaneously classifies intents and recognizes entities. Entity recognition identifies and categorizes specific pieces of information within the user's message using techniques like the DIET classifier and

Conditional Random Fields (CRF). Additionally, confidence scoring assigns probabilities to the predicted intents and entities, helping to handle ambiguous inputs by indicating the model's certainty. This robust classification framework ensures that our chatbot accurately interprets user inputs and engages in meaningful, contextually appropriate conversations.

4.0 Experimental Results and Analysis

4.1 Performance Measures

It might be said that the performance of our chatbot project is measured through various comprehensive metrics that reflect its structure and functionality. The chatbot is designed with a well-defined framework comprising 57 intents, 57 stories, and 57 utters, ensuring a wide range of user queries can be addressed effectively. The domain file, integral to the chatbot's operation, contains 235 lines of code, while the natural language understanding (NLU) file includes 676 lines, facilitating precise intent recognition and entity extraction. The stories file, which maps out user interactions, spans 284 lines. In total, the chatbot's codebase amounts to 1195 lines. These performance measures highlight the chatbot's extensive capabilities and the meticulous effort invested in its development, enabling it to handle diverse and complex user interactions with ease.

4.2 Experimental Results

The experimental results underscore the chatbot's high performance and reliability. During the evaluation at the action level, the chatbot correctly identified all 25 out of 25 actions, achieving an F1-Score, precision, and accuracy of .725 .0 These metrics indicate that the chatbot consistently delivers accurate responses, reinforcing its reliability in real-world applications. Furthermore, the model demonstrates a zero in-data fraction, implying that it is robust against overfitting and can generalize well to new inputs. Notably, when users input queries correctly in the search bar, the chatbot reliably provides the correct output, enhancing user satisfaction. Additionally, the chatbot is proficient in extracting and providing critical information such as email addresses and phone numbers, showcasing its practical utility in various scenarios. These experimental results affirm the chatbot's effectiveness and its potential as a valuable tool for efficient user interactions.

5.0 Conclusions and Future Works

In conclusion, the development of a chatbot system for the University of Jordan's Quality Assurance department addresses the pressing need for efficient and personalized communication solutions in educational institutions. Through a comprehensive analysis of the problem statement, our objective was to leverage artificial intelligence to enhance interactions between university stakeholders and the QA department, ultimately streamlining communication processes and improving user satisfaction. The outlined methodology, encompassing pipeline design, technical

implementation, dataset collection, preprocessing, feature extraction, selection, and classification, provided a structured framework for developing an effective chatbot system.

5.1 Strengths

The strength of our approach lies in its thorough needs analysis, which identified the demand for conversational agents in educational settings. By addressing project constraints such as limited data availability and technical challenges, we demonstrated resilience and adaptability in overcoming obstacles. The establishment of a robust system environment and clear software and hardware requirements ensured seamless development and deployment. Additionally, our research background and review of related works provided valuable insights into existing methodologies and technologies, informing our own approach.

5.2 Weaknesses

Despite our thorough methodology, there are areas for improvement. The limited discussion on user satisfaction metrics and error handling mechanisms in the experimental results and analysis section leaves room for further exploration. Additionally, while our dataset description and data preprocessing steps were comprehensive, a more detailed discussion on feature extraction, selection, and classification methods could enhance the understanding of our approach.

5.3 Future Works

There are a number of directions in which future work can be taken. For instance, we will make sure that our data set contains all relevant information and is accurate enough. Besides, we'll compare the performance of various classification models to identify the best ones for our chatbot system. Finally, an essential aspect of improving the model's precision is its reliability in terms of user interface interaction. In addition, if the chatbot's scope is expanded to other university sections like health services and e-learning among others, it would become more flexible and adaptable to both students' and staffs' needs. Finally, connecting the chatbot with all university platforms will lead to a seamless user experience where accessibility through any application or website associated with the institution is guaranteed.

References

[1] Python Package Index (PyPI). *pip: The Python Package Installer*. Accessed .(25 -05 -2024) 2024. URL: `https://pypi.org/project/pip/`.

[2] Inc. Anaconda. *Anaconda Distribution*. Accessed .(10 -4 -2024) URL: `https://www.anaconda.com/download`.

[3] Bobby Batacharia et al. "CONVERSE: a conversational companion." In: *Machine conversations*. Springer, 1999, pp. 205–215.

[4] Tom Bocklisch et al. "Rasa: Open source language understanding and dialogue management." In: *arXiv preprint arXiv:1712.05181* (2017).

[5] Dictionary.com. *Dictionary.com*. Accessed .(20 -5 -2024) URL: `https://www.dictionary.com/`.

[6] Rasa Technologies GmbH. *Rasa: Open Source Conversational AI*. Accessed .(4 -4 -2024) URL: `https://rasa.com/`.

[7] Ajinkya Huddar et al. "Dexter the college FAQ chatbot." In: *2020 International Conference on Convergence to Digital World-Quo Vadis (ICCDW)*. IEEE. 2020, pp. 1–5.

[8] Anran Jiao. "An intelligent chatbot system based on entity extraction using RASA NLU and neural network." In: *Journal of physics: conference series*. Vol. 1487. 1. IOP Publishing. 2020, p. 012014.

[9] Anirudh Khanna et al. "A study of today's AI through chatbots and rediscovery of machine intelligence." In: *International Journal of u-and e-Service, Science and Technology* 8.7 (2015), pp. 277–284.

[10] Khang Nhut Lam, Nam Nhat Le, and Jugal Kalita. “Building a Chatbot on a Closed Domain using RASA.” In: *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*. 2020, pp. 144–148.

[11] Mike McGrath. *Python in easy steps: Covers Python 7 .3*. In Easy Steps, 2018.

[12] Mamta Mittal et al. “Web-based chatbot for frequently asked queries (FAQ) in hospitals.” In: *Journal of Taibah University Medical Sciences* 16.5 (2021), pp. 740–746.

[13] TensorFlow. *TensorFlow: An End-to-End Open Source Machine Learning Platform*. Accessed .(10 -4 -2024) URL: `https://www.tensorflow.org/`.