Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Engineering

AI Chatbot for Students Assitance

Shirish Senthil Kumar

Supervisor: Associate Professor Kevin Kelly

April 17, 2023

A dissertation submitted in partial fulfilment
of the requirements for the degree of
BAI Mechanical and Manufacturing Engineering

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent / do not consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. **Please consult with your supervisor on this last item before agreeing, and delete if you do not consent**

Signed: ______________________ Date: ______________________

Shirish Senthil Kumar, 17 April 2023

Abstract

This paper proposes an artificial intelligence chatbot designed to address the challenge faced by the School of Engineering at Trinity College Dublin in responding to a large number of queries from prospective students. Through an extensive literature review, the chatbot was developed using the "Google Dialogflow ES" platform and a Python knowledge base curator for the FAQ section, chosen for its powerful natural language understanding capabilities. The design principles and technology development were formulated based on an extensive feasibility study, which consisted of stakeholder research via interviews, presentations, consented phone recordings, Google forms, and an in-depth SWOT analysis.

The Dialogflow agent's graphical user interface such as buttons, Trinity logos, and text hyperlinks, provides an intuitive experience to its users. Additionally, a well-defined conversational flow Is implemented to guide users and provide relevant responses. The TrinityBot was hosted on Trinity Engineering's servers and was rigorously tested to ensure its functionality and accuracy. The AI chatbot's capability was assessed using the 3-fold cross-validation, and a modified confusion matrix was constructed subsequently, to evaluate the performance of both the intent-matching methodology and the FAQ knowledge base component. The precision, recall, and F1 metrics were calculated for each methodology and analyzed to determine the overall accuracy of the AI conversational agent. The proposed TrinityBot has shown satisfactory accuracy results and a brief potential future work is recommended at the end of the paper.

Lay Abstract

A major challenge faced by the School of Engineering at Trinity College Dublin is effectively and accurately answering the high volume of questions asked by prospective students through emails, phone calls, etc promptly. These questions often have their relevant answers on the Trinity web pages, however, finding these resources can prove to be for students and newly integrated staff. A consequence of this challenge is that academic and school staff spend a significant portion of their valuable time answering basic questions. An effective solution to navigate this challenge in the form of an AI conversational chatbot **TrinityBot** is proposed in this paper.

A thorough literature review was conducted on different novel development technologies and implementation strategies with a key focus on scalability and implementation of the Trinity Engineering website. Subsequently, stakeholder research was conducted consisting of Academic Professors and Executive Officers of the School of Engineering at Trinity College Dublin. Insights from the stakeholder interviews and SWOT analysis of chatbot technology helped in the technical development of the Chatbot.

The project's development phase was divided into two primary parts. The first part involved using Google Dialogflow as a third-party tool, while the second part utilized the Python knowledge base curator to assist in integrating the AI Chatbot's FAQ section. The AI chatbot and the curator employ machine learning algorithms to analyze the data, ensuring accurate and relevant responses. The integration and embedding of the Chatbot were carried out on the Trinity Engineering Server and were thoroughly tested to ensure optimal performance and functionality. The benchmark results are discussed to support and justify the Chatbot's satisfactory accuracy.

Using TrinityBot, prospective students type in a range of questions, from general queries about the School of Engineering to more specific queries like entry requirements. It can be utilized as an added tool along with the website for students who are considering applying to the School of Engineering but have doubts about the admission process and other queries. This paper further discusses the positive impact it makes on Trinity Engineering's recruitment strategy. Students will be able to get the relevant information easily, clearing their doubts quickly and increasing the likelihood of them applying to Trinity. The possibility of the AI chatbot reducing the workload of the School's admissions team and academic professors by reducing the redundancy of basic questions and only the need to reply to complex questions is also explored. The chatbot's pre-built intents and FAQ knowledge base can be constantly updated, ensuring that it remains up-to-date with the latest information.

Acknowledgements

First and foremost, I would like to extend my deepest thanks to my supervisor, professor Kevin Kelly for his constant guidance, encouragement and invaluable technical insights throughout the course of the project. I am grateful of his trust in my abilities to successfully accomplish this dissertation.

I would also like to acknowledge the unwavering support of my family who have constantly been my backbone throughout my academic endeavours.

Finally, i would like to extend a my sincere gratitude and appreciation to all my professors and peers who I have collaborated and worked with throughout my undergraduate tenure at Trinity College Dublin.

Contents

List of Figures

List of Tables

Nomenclature

AI	Artificial Intelligence
CRM	Customer Relationship Management
SWOT	Strength, Weakness, Opportunity & Threats
API	Application programming interface
NLU	Natural Language Understanding
ES	Essentials
GDPR	General Data Protection Regulation
FAQ	Frequently Asked Questions
NLTK	Natural Language Toolkit
HTML	HyperText Markup Language
SDLC	Software development lifecycle
RBG	Rule-Based Grammar
GUI	Graphical User Interface
ML	Machine Learning
CSV	Comma Separated Value
SSML	Speech Synthesis Markup Language

1 Introduction

Artificial intelligence (AI) Chatbots have become essential to customer relationship management systems (CRM) in the modern era. Numerous industry are adopting this relatively new technology with the rise in ChatGPT-3, ChatSonic, etc. In the Healthcare industry, patients have rapid access to authenticated information and instructional material through chatbots StreamMD (StreamMD, Inc), Conversa (Conversa Health, Inc), and Memora Health (Memora Health, Inc) that run on current messaging platforms. [1]

One such industry undergoing rapid AI transformation is the tertiary higher education sector, which has seen a tremendous rise in implementing AI-enabled Chatbots. Georgia State University's "Pounce" and, Northern Virginia Community College's "Ace" are some of the AI conversational bots successfully implemented by academic institutions to increase student experience and engagement. [2] According to MainStay, formerly known as AdmitHub, chatbots can typically respond to more than 65% of applicant and student questions promptly, 24/7, and is getting more accurate and precise day by day with the help of machine learning. [3]

The implementation of chatbots has increased access to information regarding financial aid and student engagement to clarify basic queries. A chatbot can be used to effectively reduce summer melt in prospective high-school students. Due to difficulties in completing the necessary pre-enrollment chores and processes, college-bound high school graduates "melt away" over the summer and fail to enroll in college in time for the fall term and is a prevalent issue amongst prospective students. [4] As a result of the deployment of the Pounce Chatbot at Georgia University, there was an overall increase in enrollment by 3.9% and a decrease of summer melt by 21.4%, captured by Dimitriadis in [5].

This technology is very beneficial for international students as they bring in numerous queries. The process of living in a different country can be quite overwhelming and clearing their queries can reduce their anxiety of taking admission at Trinity College Dublin. At Iuliu Hațieganu University of Medicine and Pharmacy, international students brought in 3,600,000 Euros or 76% of the total annual tuition fees collected in 2020. There is a need to ensure effective interpersonal communication with the international student community. [3]

With the digital transformation of different institutions and the transition from conventional college admission to automated processes, an AI chatbot can be beneficial for young prospective students. Furthermore, Socially anxious customers desire anthropomorphic customer service chatbots and conversational agents rather than direct communication with a human being. [6] According to research, the younger generation prefers messaging apps like Facebook Messenger, WhatsApp, and SMS for communication over phone calls or other direct person-to-person means. [7]

Gartner research predicted that by 2020, the average Facebook Messenger user would speak to bots more frequently than partners daily and that 85% of client-brand engagements would not include direct human contact. Furthermore, greater than 50% large and mid-sized international businesses would install chatbot-based solutions, primarily conversational ones, around the same time frame. According to Grand View Research, this would drive the worldwide chatbot market value to $1.23 billion by 2025. [8]

1.1 History

The general concept of simulating human speech in the form of text messaging using artificial intelligence (AI) methodologies dates back to the 1950s when the renowned Alan Turing—best known for his work as a cryptanalyst during World War II—developed the Turing machine. [9] He essentially proposed the question "Can machines think?", which intrigued computer scientists around the world to emulate human intelligence in computers and machines.

Here is where chatbots entered the scene, as a utility software, advisor, or simply a friend you may converse with. According to the lexicon, a chatbot is defined as 'A computer program designed to simulate conversation with human users, especially over the Internet'. [10] In more technical terms, the chatbot employs Natural Language Processing (NLP) and sentiment analysis to text or speak in human language with people or other chatbots. [11]

In 1960's, the first chatterbot "ELIZA" was developed by Massachusetts Institute of Technology (MIT) which used pattern matching to give canned or pre-recorded responses, giving a very basic sense of human interaction to the user. However since the chatterbot was developed in early stages of Artificial Intelligence, the system was frail but passed a restricted Turing test for machine intelligence. [12]

Following the dot-com bubble, companies and universities developed the first industrial chatbots that incorporated natural language understanding algorithms with intuitive graphical user interface. Examples include: HEXBOT developed in 2004 and Artificial Linguistic Internet Computer Entity (ALICE) developed by Dr. Richard Wallace in 2002 using Artificial Intelligence Markup Language (AIML). [13]

With further technological advances in recent years and progress in machine learning algorithms and data mining methodologies, AI Conversational bots in the form of personal assistants have been integrated with various devices and websites. These agents most often have a conversational flow and engage in active conversation to deliver a seamless experience to the user.

Multinational technology firms have invested in and developed advanced AI conversational agents to answer the queries asked by users automatically, on a much larger scale, which in turn reduces labor costs. [3] Some examples of modern conversational agents include: CONVERSE by salesforce research [14], Alexa virtual assistant from Amazon, Siri personal assistant from Apple ecosystem, and most recently ChatGPT-4 from OpenAI. These voice-activated virtual personal assistants (VPAs) have been beneficial for users with cognitive disabilities and imparities while enforcing inclusivity by recognizing accents of different races. [15]

Artificial Intelligence is growing every year and is said to be outpacing Moore's Law, which states that the number of transistors on a microchip doubles every two years. In other words, AI is growing multifold and is growing at a pace faster than the semiconductor chip industry. [16]

1.2 Objectives

This paper highlights the challenges faced by the School of Engineering at Trinity College Dublin in effectively and accurately answering the questions from prospective students and proposes a novel methodology through the implementation of an Artificial Intelligence Chatbot "TrinityBot". The measures utilized to determine the practicality and usefulness of this project involve a thorough feasibility study through stakeholder research, a literature review of different integrating platforms, and an effective research proposal.

Firstly, an initial SWOT analysis and stakeholder research were conducted. The contributors for this research comprised of prospective students, school officers, technical experts and academic professors in Trinity Engineering and the insights gathered are used to refine the designing strategy. Furthermore, this allowed us to define the scope of the project as well. The objective is to develop a robust, functional and functional AI Chatbot for prospective students looking to join the School of Engineering at Trinity, all within a narrow time frame. Through the results of the research, ideal design principles for the chatbot were concluded that is in accordance with the admissions strategy at Trinity College Dublin.

The Chatbot agent implemented acts as a conversational inquiry management tool and assists prospective students in their admission process. It is intended to answer frequently asked questions (FAQs) effectively and accurately while providing a great degree of human parity. The agent uses artificial intelligence and machine learning algorithms to give relevant answers and hyperlink specific information and web pages in Trinity Engineering's website based on student queries. The intuitive appeal of the graphical user interface is equally important for the success of this project.

A research proposal was devised from a technical literature review of different technologies. The literature review comprised journals, books, and internet websites to select the appropriate programming language and integrated platform. For this project, Python programming language and its libraries, mainly NLTK (Natural Language Toolkit), PyTorch, and NumPy were used to create a data curator that assisted in creating the knowledge base of the chatbot. This was integrated as the Google Dialogflow Essentials (ES) FAQ section in a ".csv" file format. In addition, custom responses that have been coded have also been used to create the conversational agent to provide a user-friendly experience for the user. These features were scaled and integrated with the Trinity Engineering Website.

Testing and Benchmarking of the chatbot was performed to validate the ability of the agent to deliver the intended results with good precision and accuracy. The initial course of action was to perform the testing and analysis was the construction of a confusion matrix as the chatbot uses intent classification to match with an appropriate response. Using the confusion matrix, the precision, and the recall, an F1 classification score was calculated accordingly. All the results were recorded manually in a Microsoft Excel spreadsheet.

2 Literature Review

The literature review conducted comprised of journals, books, and existing technologies, the principle of intent matching was chosen for the conversational agent to respond accurately. Since Artificial Intelligence and Chatbots are relatively new technologies, research is constantly conducted to innovate novel algorithms and efficient machine learning techniques.

2.1 General Methodology

A typical approach for developing AI chatbots involves three main steps: intent classification, intent matching and generating a response using pre-built code associated with the matched intent. The working architecture of the Chatbot and the algorithms that were utilized to make an effective match was explored in this section. The key definitions and purposes of components of the agent such as intents, entities, training phrases, validation, sentiment analysis and confidence scores are explained to contextualise the different technical processes that occur in the backend of the AI Chatbot.

2.1.1 Working Scheme of an AI Conversational Agent

Figure 2.1 represents a general chatbot session where initially, the end-user inputs queries that are processed and classified on intents. [17] An intent represents the possible user's intentions and is typically reflective of the use cases of the chatbot, in this project, information regarding Trinity Engineering for prospective students. [18] The intent classification is a vital part of the natural language understanding pipeline for conversational agents and helps attach an intent confidence score (0-1) to the query. [19] The closer the score is to 1, the more confident that the chatbot is in assigning a particular intent. This in turn relays the appropriate predefined response back to the end-user via an messaging interface and hence one chatbot loop/session is completed. A lifespan can be set for the chatbot which represents the number of times a query can be parsed in the chatbot, based on server capacity. The external services include the prospective student's database that the chatbot accesses to reply to the end-user. [18]

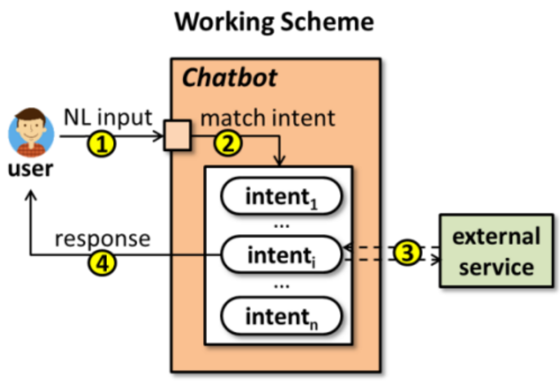

Figure 2.1: Working scheme of an AI Chatbot [18]

2.1.2 Technical Strategy

The technical development of the AI chatbot took place in two stages that ran simultaneously. One stage was the development of the AI chatbot using a third-party tool "Google Dialogflow Essentials (ES)". Although there are numerous conversational agent development tools in the market such as Watson by IBM, LUIS by Microsoft, and Lex by Amazon, there are certain benefits in using Dialogflow over other market tools.

Dialogflow ES is a standard conversational agent development tool that is free and more importantly, negates the issue of integration and scalability by providing the feature of application programming interface (APIs). [20]

The application is relatively easy to work with and provides deep control of the conversational flow of the chatbot. [18] In this project, the chatbot delivers the FAQ section with predefined responses (canned answers) and a section for custom responses with rich response messages delivered based on the matched intent. However, one drawback was that, the integration and testing of the FAQ section (Canned Responses) was difficult as it required a constant back and forth between different tabs.

To address this issue, the second stage was introduced to the project in the form of a Python AI chatbot Data Curator. This Python program assists in curating the FAQ section (knowledge base) of the Dialogflow conversation agent by iterative feedback looping. Feedback looping is an AI strategy to refine and improve the data, to help the chatbot get more accurate. [21] It also gives a basic understanding of the natural language understanding (NLU) processes in the background that help the chatbot match intents to give relevant responses. Both the prototype and Dialogflow follow a similar principle and methodology as shown in figure 2.1. The objective of this data curator is to provide a rich, well-defined, and strong knowledge base that can be easily integrated with Google Dialogflow in the form of a ".csv" file. A basic yet effective GUI is coded to the prototype for ease of working and data manipulation.

2.1.3 Comparing Conversational NLU Tools

The conversational NLU tools have different perks and is evaluated through literature review to select the most appropriate chatbot. The four main NLUs in the market are Rasa, IBM Watson, Google Dialogflow and Microsoft LUIS and is compared in this section. Since the chatbot implemented runs on the methodology of Intent Classification, the most effective evaluation metric to determine the accuracy of the chatbot is based on F1 score, that is constructed using a confusion matrix. This score is a machine learning metric that combines the precision and recall to calculate the F1 score. The score ranges from 0.0-1.0 and is primarily used in binary classification, in this case, "yes" if the intent is matched and "no" if not. [22] In 2.2, Dialogflow performs well with an accuracy of 83.1%, giving an accuracy of 0.831. [23] This is relatively better than Rasa and LUIS, matching intents more accurately and providing the intended responses. However

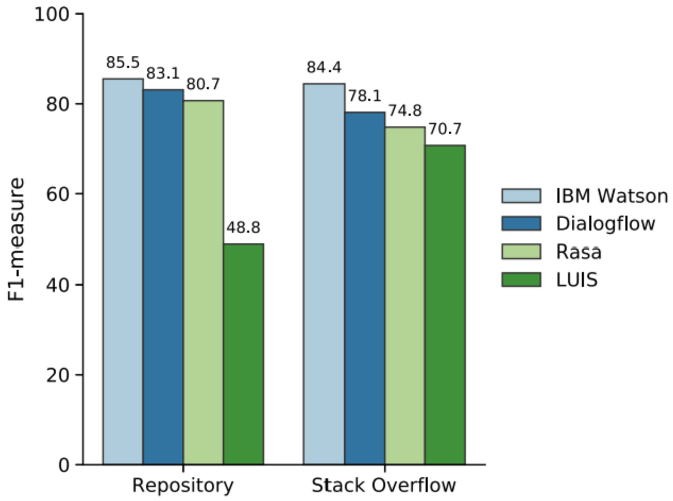

Figure 2.2: Intent classification performance as F1-measure of the four NLUs. [23]

An important feature of Dialogflow is sentiment analysis which is not available in other conversational development platforms. [24]

2.2 Google Dialogflow

Dialogflow also provides speech recognition which serves as an effective tool for the disabled community with visual impairment. Google Dialogflow ES enables the developer to have a maximum of 2000 intents with a maximum of 2000 training phrases per intent. [25] The different components and methodologies incorporated in the Google Dialogflow conversational agent builder are Intents, Entities, Training Phrases, Knowledge Base, Integration, Validation, and Analytics. These components are best explained through examples therefore technical literature is explained using a weather chatbot.

2.2.1 Intents

Intents are quite simply the user's end-users intentions to use the chatbot. In Trinitybot, these intentions could be queries regarding the available courses, job prospectus, scholarships, etc. In 2.3, The words highlighted in blue ("Forecast", "Weather", "Temperature") show the end users' intentions and the conversational agents match it to the forecast intent.

Figure 2.3: Intent Classification [20]

2.2.2 Entities

An entity is defined as the specific information or categorical data that can be extracted from the query inputted from the end user. [24] The predefined data types or entities can be addresses, cities, etc. [26] In addition, custom entities can also be created for specific usage of the chatbot. A benefit of having entities is that it helps in categorizing content based on the chatbot preferences and staying organized. [27]

2.2.3 Events

When an end-user expression fits an intent training phrase, the intents are matched. But, events can also be used to trigger intents. Custom events have been set throughout the TrinityBot, to trigger the intents in chronological order. This gives better control when the end-user is to be filtered from the main branch of the conversational flow to a side branch. [28]

2.2.4 Sentiment Analysis and Confidence Score

As mentioned before, one added feature of Google Dialogflow is the use of sentiment analysis. The feature tries to identify any subjective opinions attached to the query entered by the user. [27] Based on the query, a normalized sentiment recognition score ranging from -1.0 to 1.0 is calculated. Here, -1.0 refers to an extremely negative emotion and 1.0 is a positive emotion, while neutral is 0.0. [24] For the trinity bot, sentiment analysis can be harnessed to escalate messages that have a high negative emotion directly to a human that is in the School of Engineering to clarify the prospective student's query and can avoid frustration. Currently, sentiment analysis only works in English and text-based queries since they do not consider the audio signal's characteristics. [24]

When the chatbot successfully categorizes intentions based on the question supplied by the end-user, it is determined by the AI chatbot confidence score. The scale goes from 0 to 1, with 1 denoting a high degree of confidence that the right intent was understood and vice versa. The degree to which the pre-built training phrases resemble the user-entered query determines the score primarily. In Dialogflow's settings, a confidence threshold can be defined, meaning that if the score falls below the threshold, the default fallback intent is activated. This is especially helpful when it comes to identifying and validating the AI chatbot. [23]

2.2.5 Training Phrases, Actions, and Parameters

Special attention to detail is given to what actions and training phrases might trigger an intent match. In an AI chatbot, these training phrases are added before testing and have an element of prediction of what an end-user might ask for a particular intent. The training phrases suggest a few phrases as to what the end-user might prompt. It is important to create the Natural Language Understanding (NLU) model that allows the chatbot to recognize the intent, and extract the entities and parameters that are in relation to the query. [24] Once there is a similarity between the training phrase and the query the user entered, the intent is matched and the action is triggered. [20] This action is essentially the extraction of parameters and their usage in the response accordingly.

In figure 2.4, the conversational agent engages in weather forecasts with the end user. For example, if the user enters the phrase "What is the weather in Bangalore on Monday?", based on the training phrases, the weather intent is matched by NLU, and the two parameters namely "Date" and "Location" are extracted from the query. The word "Monday" is in relation to a particular day/date and "Bangalore" to a city. Furthermore, the preset entities "@sys.date" and "@sys.get-city" are searched and if there is a match with the extracted parameter, it assigns it as values to "$date" and "$location". In other words, Date = Monday in @sys.date and City = Bangalore in @sys.geo-city for this chatbot instance. The values in figure 2.4, "$date" and "$location" can be used to customize the response accordingly. An example of an appropriate response for this query would be "The weather in Bangalore today is 25° Celsius. Dialogflow understands a certain pattern with the addition of more training phrases.

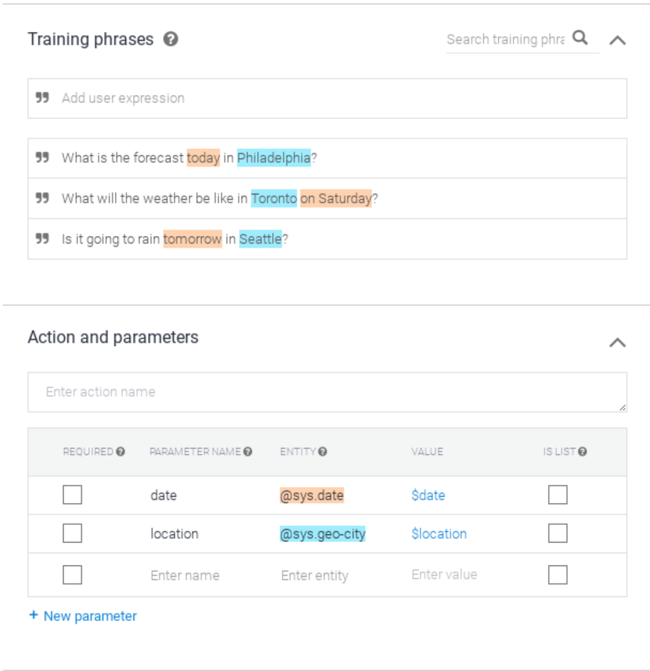

Figure 2.4: Training phrases and actions in Google Dialogflow [29]

2.2.6 Algorithms

Dialogflow relies on two important algorithms to match intents accurately. These two algorithms are Rule-based grammar matching and Machine Learning (ML) Matching. [30] Dialogflow defaults to attempting both of these algorithms simultaneously and selecting the more accurate outcome. If a match is not made using RBG matchmaking, it switches to ML matching. [26] This results in an optimized way of finding a solution.

- *Rule-Based Grammar Matching*: RBG matchmaking methodology enables the chatbot to function according to a set of rules. The end-users and the developer's language abilities are both factors in the Rule-Based Grammar for the Trinity chatbot. It considers the grammar's minute details before contrasting them with the predetermined intents and generated intents from a chatbot session. If the intents are both comparable and similar, then the chatbot links both intents, and the response is triggered. [26]

- *Machine Learning*: There are a plethora of ML algorithms that Dialogflow matching-based technique implements for mapping the intents to the required action. Although it is not clear as to which algorithm is used, one key takeaway is that it does not function accurately if there are less training phrases and small data sets. [26] The ML matching can be toggled off if required in the Dialogflow settings, however this is not advisable as it lowers accuracy. [27]

Rule-based grammar matchmaking is preferred for faster training of phrases and more accurate responses for chatbots with smaller datasets. However, a combination of both RBG and ML algorithms was essential for the Trinity chatbot as the knowledge base and number of intents were considerably large. This gave it the capability to self-learn and get more accurate over time as more predefined intents were entered. [31].

2.2.7 Input and Output Contexts

Contexts are vital to any chatbot as it provides conversational control to the developers. For each intent, it consists of the input and output contexts. [26] The developer can configure the chatbot by linking the output context of an intent, to the input context of a different intent. In figure 2.5, the AI chatbot prompts the user "What information do you need for your checking account?", to which the user enters "My balance". The chatbot, redirects this conversational flow, based on intent matching to the "CheckingBalance" intent which responds with the user's checking account balance.

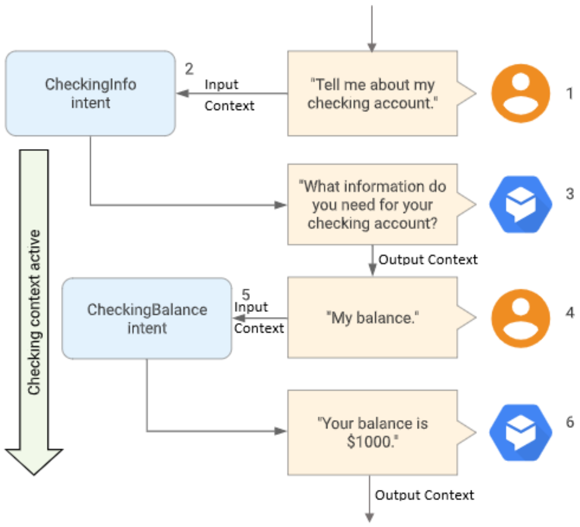

Figure 2.5: Conversational flow representation between two intents using contexts. Adapted from: [32]

2.2.8 Knowledge Connectors

Dialogflow enables the chatbot to answer repetitive questions by having a dedicated knowledge base. [33] When none of the intents match, the knowledge base is searched. In addition, the preference is adjustable in Dialogflow allowing the chatbot to either match via intents or knowledge base. It also effectively eliminates the need for manually defining different intents and their responses. [33] However, one disadvantage of using this connector is that it gives standard responses without any possibility of integrating rich text messages (hyperlinks, buttons, etc). As mentioned in the technical strategy, the knowledge base connector acts as the FAQ section for the TrinityBot. Another major drawback was the difficulty in testing the data entered into the knowledge base. Therefore, the Python prototype was built to curate the FAQ section data and can extract as a ".CSV" file for direct implementation in the builder. If the FAQ section created using the Python Prototype basic NLU adequately addresses the questions pertaining to prospective Trinity Engineering students, it will be more accurate when used with Google's robust NLU engine.

2.2.9 Agent Validation

The validation mechanism is a method of rectification in the event that Google Dialogflow does not match the intent although to the developer the entered query is similar to the training phrases. The method of validation is automatic, however it can be done manually by forcefully assigning the intent that was supposed to be matched. Dialogflow also highlights this error automatically and enables the developers to do the forceful validation. Therefore, when the same query is asked again, the chatbot after validation, matches it with the appropriate intent. [34]

2.2.10 Responses, Integration, and Embedding

The responses are in two categories, mainly as "Text or SSML" or a "Custom Payload". Text responses refer to plain text responses to the end user. Custom Payloads allow for rich text responses and the developer can code in pictures, logos, hyperlinked buttons, etc to make this intuitive to the end user.

The Google Dialogflow conversational builder is very versatile and can be integrated into different platforms. These platforms include Facebook, Slack, Messenger, Twitter, etc. [35] The end-user interactions are monitored and effectively handled by the Google servers, making it easier for scalability and integration. [36] It usually involves plugging in the code snippet in the platform or the website source code. For TrinityBot, the feature "Google Messenger" is used for its simplicity and ease of implementation on a website. The ability to send rich text messages makes this the ideal integration option, as it makes use of rich graphical user interface features.

The Trinity Engineering Website is hosted by the School of Engineering Department and developed through Adobe Dreamweaver as a PHP server. The benefit of using Dialogflow Google Messenger is the ease of integrating this conversational agent into the PHP server. The developer only needs to enter the link to the javascript file that contains the chatbot, through the script source. The website accesses the conversational agent that is run on the Google Servers, so there is minimal server load to Trinity servers.

2.2.11 Combined Technical Architecture of Dialogflow Agent

The whole technological framework of Dialogflow and the interactions between its various components are shown in Fig 2.6. The four blocks of the AI Chatbot's diagram are Embedding, Intent, Query, and Response. The query block is initially responsible for receiving the end-user's query and extracting parameters. The agent's intent categorization and matching processes are handled by the intent block. It contains the entities and pre-built training phrases that aid in constructing the confidence score when the chatbot tries to intent match. With the help of the rich text message characteristics, the answer block in the figure gives the end-user the desired response. The embedding block allows for deployment in different social platforms or in this project, the Trinity Engineering website.

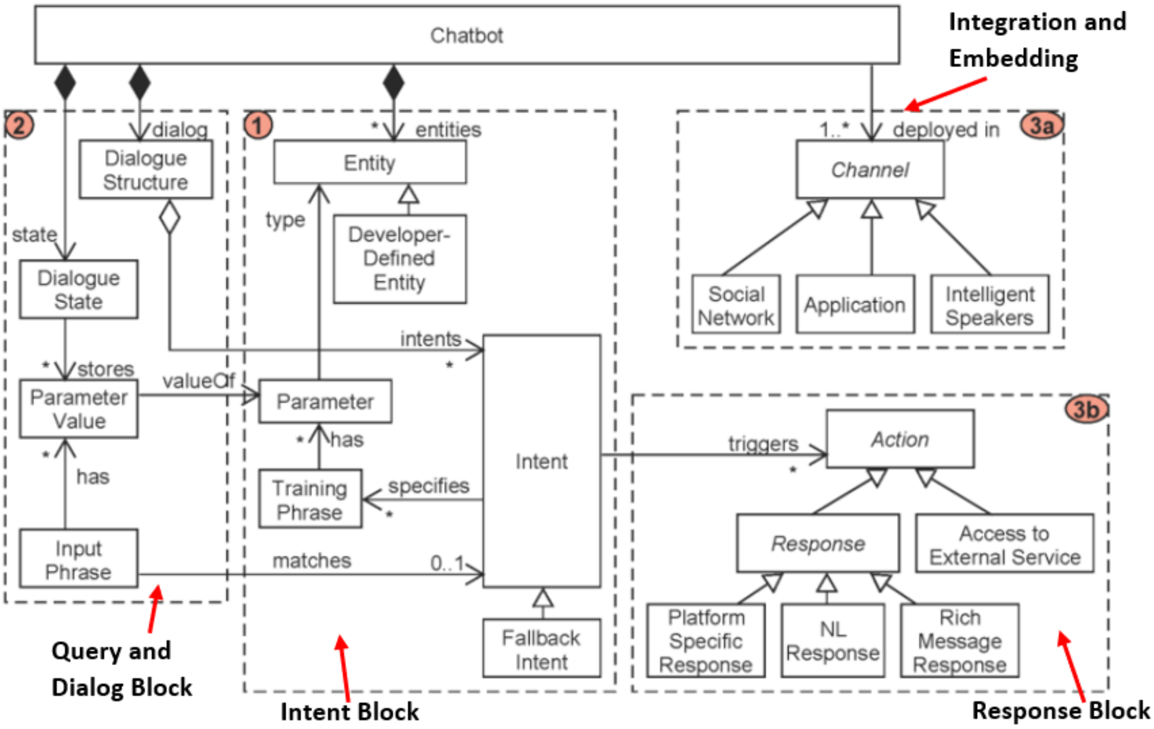

Figure 2.6: Overall Technical Architecture of Google Dialogflow adapted from: [18]

2.2.12 Challenges Associated with Google Dialogflow

Initially, a major challenge that was hypothesized with TrinityBot was a failure to integrate the chatbot embedding code to the PHP servers in Trinity College Dublin. Following a literature review and successful implementation, this technical risk was no longer an issue. Since the conversational agent interacts with third-party servers in this case Google, there are possible risks in relation to GDPR and Critical Data Leakage. Furthermore, there is a risk of misusing this data like the sale of Trinity student data to technology firms. [37] It is ideal for saving the data in the cloud to enable data recovery in the event of data loss.

2.3 FAQ Knowledge-Base Curator

The Python curator built is a fully conversational agent with a basic Natural Language Understanding, built to assist in refining the knowledge base that is integrated as the FAQ section in Google Dialogflow. This can be integrated with the "knowledge connector" as mentioned in the previous section through a ".CSV" file format. The curator is built with basic neural network principles and is based on the literature review conducted on [38]. The code and program also provide a brief blueprint of the complex processes in the Google Dialogflow NLU engine. The Python program is coded in the PyCharm environment.

2.3.1 General Methodology of the Curator

Through literature review, an effective data-curating prototype was built that runs on intent classification and matching, the same principle as Google Dialogflow. Fig 2.7 shows the methodology and flowchart involved in developing this tool. It can be seen that methodology is similar to Dialogflow NLU, however, the pre-defined responses can be trained and tested for coverage, much faster than Dialogflow. Having the program side by side with the intents.json file and testing saves a lot of time.

When a query is asked by the user and the intended response is not matched by the NLU, the default intent is triggered which indicates to the developer that the knowledge base has to be tweaked to cover that query. This is The training phrases extracted from this curator, were entered into a separate Excel Spreadsheet, and the ".CSV" file was extracted. This file can directly be integrated into the Dialogflow knowledge base connector.

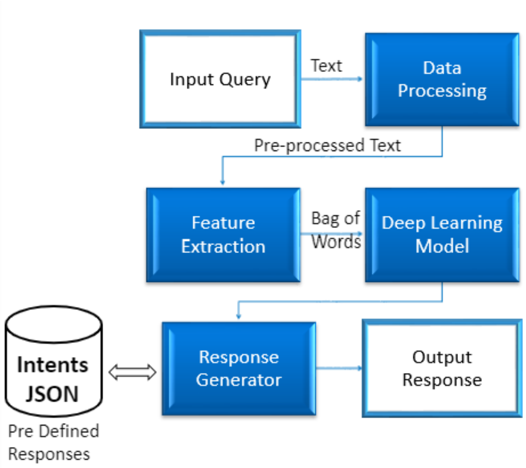

Figure 2.7: Python Knowledge Data Curating Flowchart

2.3.2 Python Scripts and Modules incorporated

The program consists of 5 main files to emulate the functionality of Dialogflow. These files include "train.py", "chat.py", "model.py", "nltkutils.py", "Intents.json" and "App.py" and all serve different purposes in the curator.

- intents.json: This file contains the intents the curator classified and matches, which is similar to the intents in Google Dialogflow. [38] All the FAQ Data is entered in this file and tested. Defining the intent keeps track of what topics have been entered, in relation to prospective education. The JSON file consists of tags (intents), patterns (FAQ), and responses as seen in fig 2.8.

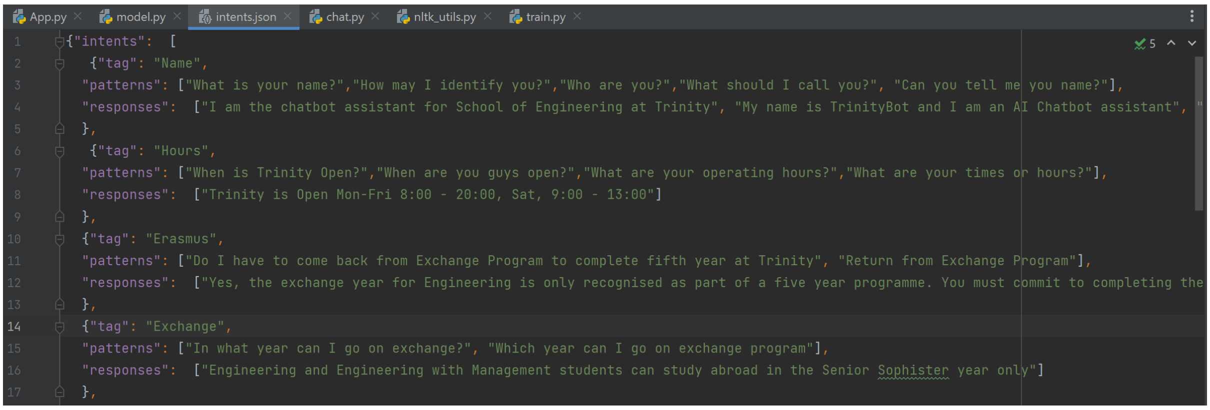

Figure 2.8: JSON file containing Intents in TrinityBot

- Model.py: The code for the neural network model can be found in "model.py." To create a neural network with PyTorch, a stable version of PyTorch is needed Torch and Torch.NN modules are installed. The class NeuralNet extends the module Torch.NN, taking into account the input size, hidden size, and the number of different tags/intents. The neural network comprises of three layers: input, hidden, and output. The input nodes correspond to distinct words taken from queried phrases, while the output nodes correspond to different tags/intents. In FigA1.4 Rectified Linear Unit (ReLU) activation was applied between layers, which produces the output node. Thus, if it's positive it returns 1 and 0 if there is no match. Furthermore, the SoftMax function is used on the output layer for classification purposes. [38] The code implemented can be found in the appendix A1.4.

- NLTK_utils (Natural Language Toolkit): The script employs the NLTK library to perform input stemming, such as reducing words like "programming," "programmer," and "programs" to the common term "program." The NLTK library is used as a toolbox for computational linguistics in Python. In appendix Fig A1.5, the "Tokenize" function works with words or sentences. Using the nltk.word_tokenize built-in function was utilized to divide incoming sentences into words. The stem function reduces a word to its stem. The process of tokenizing and stemming results in a collection of words referred to as a "bag of words". This bag of words is then used as input node for the neural network module. [39] The code can be accessed in the appendix A1.5.

- Train.py: From model.py and nltk_utils.py, the functions involving tokenization, stemming, and removal of duplicate words are called. The data is extracted from the "intents.json" file. This data was then formatted and saved in a new class named ChatDataset. Through the literature review, the hyperparameters were determined for this machine learning model. A hyperparameter in machine learning is a preset parameter that influences model training and is defined by the user before training. In Fig 2.9, the selected hyperparameters (epochs, batch size, and learning rate) for Trinity Knowledge Data Curator are ideal to give great accuracy. [40] An object of the NeuraNet class was created with the selected hyperparameters, and the dataset was loaded using the DataLoader function from the Torch library. Depending on availability, the model was trained using either a CPU or GPU on the local computer. The CrossEntropyLoss function was employed for calculating loss and the learning rate was used for optimizing the algorithm used. Now the curator is trained and ready for use. [38]

```
# Hyper-parameters
num_epochs = 1000
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
```

Figure 2.9: Selected Hyperparameters for the Chatbot [40]

- Chat.py: This Python code encapsulates all the scripts mentioned above such as Neural Network Model, Natural language understanding, training, and optimal hyperparameters. It is responsible for parsing information and liaising with the modules mentioned above to get the curator's responses. It is further involved in passing this response to App.py for the program to display with the simple GUI. Finally, if there is no match, it also provides the default intent response "I do not understand" to indicate that the query is not covered by the FAQ section, therefore.

- App.py: This Python script utilizes Tkinter Library to provide an effective and simple GUI for the application. The library comes with numerous functionalities including a message entry box, send button, demarcation of the user's queries, and the response of the data curator. This file is run to see the final GUI, that is used for the curating process and this can be seen in Fig 2.10. [38]

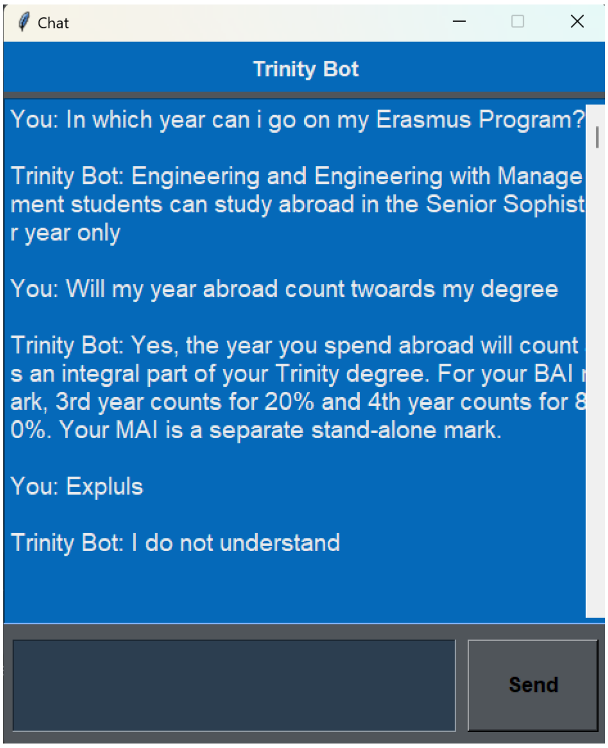

Figure 2.10: The GUI for TrinityBot Curator

2.3.3 Challenges Associated with the Data Curator

As mentioned before, the idea of implementing the basic NLU Python data curator as the conversational agent of Trinity Engineering was explored. However, a major drawback is the scalability of this prototype as embedding this code into the Trinity Server and other platforms requires extensive coding. In comparison to Google Dialogflow which embeds the conversational agent through a <script> HTML tag and the URL of the chatbot. [28] The data curator can be optimized with a stronger natural language understanding to provide a richer knowledge base. The challenge of the General Data Protection Regulation (GDPR)is a risk that is contained with proper storage of data on the cloud.

2.4 Testing Methodology and Evaluation Metrics

The chatbot is tested thoroughly in order to prevent it from failing in any scenario and to predict the accuracy of the Artificial Intelligence Chatbot. In this project, it is vital to test the intent-matching capability and the FAQ knowledge base. For intent matching, the K-fold cross-validation is a favorable technique. Additonally, using this technique, the confusion matrix is constructed on the basis of the correct responses of the chatbot. The confusion matrix is a tool that is commonly used to evaluate the performance of machine learning algorithms. [19]

However, the confusion matrix created is modified to account for the implementation of a default fallback intent. [41] The subsequent data recorded is used to calculate the precision, recall, and F1-score. [19]

2.4.1 K Fold Cross-Validation for Intent Matching

There are several algorithm inspection techniques and among these techniques, K-Fold cross-validation is a popular approach as intent classification and matching are important methodologies for the Chatbot in this project. [42] The K-Fold cross-validation is usually implemented on artificial intelligence systems that involve a classification problem. In this technique, the training dataset is divided into K segments of the same sizes for K iterations. During each iteration, a portion of the data is set aside for validation while the remaining K-1 splits are utilized as the training dataset. [43] The figure 2.11 represents how the cross-validation takes place for different iterations. The K-Fold Cross-validation is performed and using this technique, the modified confusion matrix is constructed for Intent matching. This data is recorded in Microsoft Excel.

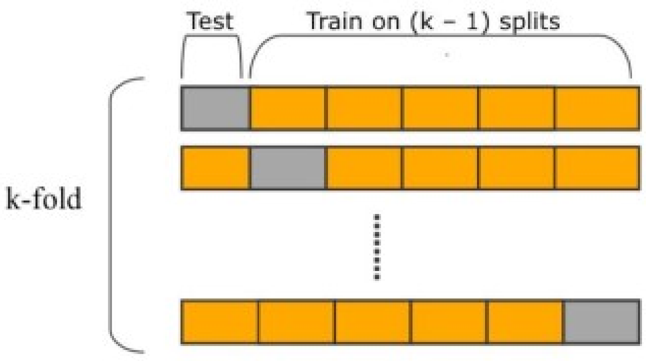

Figure 2.11: Methodology of K-Fold Cross-Validation [43]

2.4.2 Confusion Matrix

Measuring the precision, recall and F1-score of the intent-matching methodology and FAQ section is done by constructing a confusion matrix. However, in the case of TrinityBot consisting of the default fallback intent, the traditional confusion matrix may not be appropriate since it assumes that the chatbot will always provide a response. Instead, a modified confusion matrix can be used that takes into account the default fallback intent. The modified confusion matrix has the following elements:

- True Positive (TP): The chatbot correctly identified and responded to the user's intent.
- False Positive (FP): The chatbot incorrectly identified the user's intent and responded with an irrelevant and different intent. For example, a query based on scholarship

matches the intent of the course. During testing in the FAQ section, a false positive occurs when the system provides incorrect answers to questions from the knowledge base.

- True Negative (TN): The chatbot correctly identified that the user's message could not be catergorized/classified and responded with the default fallback intent.
- False Negative (FN): The chatbot incorrectly identified the user's message as unclassifiable and responded with the default fallback intent, but the message could have been classified as a specific intent. Using these four metrics, you can calculate the precision and recall of the chatbot and further attach an F-1 score. [44]

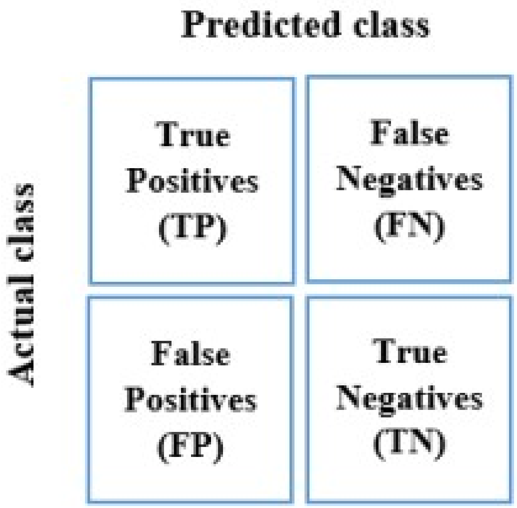

Figure 2.12: Confusion Matrix based on responses [43]

Figure 2.12 shows the confusion matrix that can be built using the positive and negative values generated from cross-validation. using this modified confusion matrix, the performance of the chatbot can be evaluated, taking into account the fallback intent, that is triggered if there is no intent match. For this project, the confusion matrix has been represented in a tabular form to for easier visualization.

2.4.3 FAQ Knowledge Base Testing

The testing of the FAQ knowledge base layer is done by querying the chatbot in separate iterations. The chatbot is presented with queries that closely resemble those pre-built in the knowledge base, and its response is recorded. The modified confusion matrix is calculated and precision and recall are obtained. Subsequently, the F-1 score is calculated which gives the accuracy of the FAQ section.

2.4.4 Precision, Recall and F1-score

In this project the evaluation metrics are calculated for the intent matching and FAQ section of the AI Chatbot. Precision measures the proportion of correct responses among all the responses provided by the chatbot and is shown in equation 2.1. Recall measures the proportion of correct responses among all the possible correct responses and is calculated using equation 2.2 Using the recall and precision, the F1-score is calculated which is represented by equation 2.3. This F1-Score ranges from (0-1) and gives the accuracy of the intent matching and FAQ section of the agent. The closer the score is to 1, the more accurate is the methodology and system. The estimated F1-score as shown in Fig 2.2 for Google Dialogflow is **0.83**, which is very accurate for a machine learning model. Through testing, a score of **0.83** would result in a functional AI conversational agent.

- *Precision of the AI Chatbot*: [43]

$$\mathit{Precision} = \frac{\mathit{True\ Positive\ (TP)}}{\mathit{True\ Positive\ (TP)} + \mathit{False\ Positive\ (FP)}} \tag{2.1}$$

- Recall of the AI Chatbot: [43]

$$\mathit{Recall} = \frac{\mathit{True\ Positive\ (TP)}}{\mathit{True\ Positive\ (TP)} + \mathit{False\ Negative\ (FN)}} \tag{2.2}$$

- F1-Score of the AI Chatbot: [19]

$$\mathit{F1\text{-}Score} = \frac{\mathit{Precision} + \mathit{Recall}}{\mathit{2}} \tag{2.3}$$

3 Feasibility Study

3.1 Stakeholder Research

The stakeholder analysis was vital as the key participants had a direct influence in the decision-making process for the AI conversational agent. Their insights helped narrow down the real issues and helped build an effective chatbot solution that benefits all the parties. This is one part of the feasibility study and the key takeaways from this study will decide if a chatbot is actually practical, realistic, and an effective solution that is worthwhile pursuing. For this project, both subjective opinions and quantitative data are equally important to conduct the feasibility study.

3.1.1 The Stakeholder Participants and Methodology

The participants comprised prospective students, school officers, technical experts, and academic professors. Some notable stakeholders involved in this project are Dr. Kevin Kelly (Associate Professor, Mechanical, Manufacturing & Biomedical Eng), Dr. Declan O'Loughlin (Assistant Professor, Electronic & Elect. Engineering), Dr. David William O'Connell (Assistant Professor (Civil Struct & Env. Eng) and Miss Sarah O'Brien (Executive Officer, School Office - Engineering), The academic stakeholders, mainly the school officers and professors were shown a brief Microsoft Presentation about the initial proposal of the chatbot technology and highlighted some of its pros and cons. This was followed by a structured and consented interview with a questionnaire. Additionally, a follow-up Google form survey was circulated amongst the technical department and school office to get more professional opinions on this technology.

3.2 Analysis and Key Takeaways

In total, 7 responses were recorded (4 from the school office and 3 from the technical department). The survey has a combination of objective and subjective questions as both are important to this project. The responses were recorded in audio, google forms and spreadsheets.

3.2.1 Subjective analysis

Taking the Google form data, the stakeholders key takeaways have been taken into consideration. The following questions were asked to the stakeholders in the form of a Likert scale (disagree to agree).

- *Q1:* In your opinion, do you agree or disagree with the following statement: An FAQ Chatbot would reduce the number of questions and save the overall time spent on prospective student enquiries.
- *Q2:* In your opinion, do you agree or disagree with the following statement: A structured FAQ Chatbot as an added tool to the Trinity webpages would fit in the overall admissions strategy of the School of Engineering, Trinity.
- *Q3:* In your opinion, do you agree or disagree with the following statement: You are confident that a chatbot can improve the efficiency of handling admissions inquiries?
- *Q4:* In your opinion, do you agree or disagree with the following statement: An FAQ Chatbot that can be maintained and updated by school officers could be a technology explored by the School of Engineering at Trinity in the future.

The stacked bar chart in Fig 3.1 presents a summary of the Likert scale data based on the responses of the 7 participants. The chart highlights their opinions regarding the use of an AI chatbot at Trinity for various purposes.

For the first question, most stakeholders agree that an AI chatbot could have the potential to reduce the number of questions asked to academic and school staff. However, a signification number of the participants are either undecided or disagree with the idea that a chatbot designed for Trinity could capture the admission strategy implemented at the school of engineering.

When asked about the efficiency of an AI chatbot in assisting the admissions team with prospective students, the majority of the stakeholders express a strong positive response. Finally, the stakeholders' opinions are mixed when asked if Trinity as an institution would explore AI Chatbot technology in the future, but the majority are optimistic and in favor of such an initiative.

There are considerations when developing the design principles for the AI chatbot highlighted by the participants. There are doubts and concerns regarding the Chatbot's effectiveness and accuracy while arguing that it might result in an increase in the number of questions asked instead of reducing them. They also believe that it would be very impersonal when talking to the agent.

However, some stakeholders have proposed that the AI chatbot could serve as a tool to identify information gaps on the Trinity Engineering website while interacting with students.

Some paraphrased quotes from the Google form survey:

- If the Trinity Engineering website fails to address the standard inquiries posed by students, it indicates a major flaw in the website. Also, when it comes to complex queries, a chatbot can find it difficult to respond, which can only irritate users further. This can be both impersonal and annoying to end-users.
- Asking a chatbot for help in finding information on an MSc course, while already on the department website, suggests a real issue with the website. Improving the website's functionality and fixing any missing elements would be a better use of this tool.

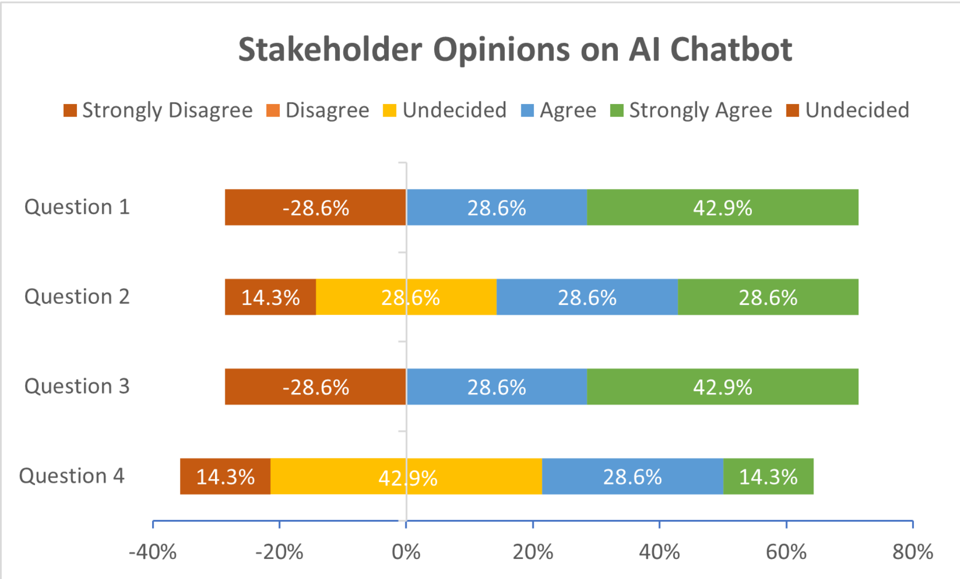

Figure 3.1: Stakeholder subjective question responses

3.2.2 Quantitative Analysis

In Fig 3.2, the frequency of questions based on the intent of questions was recorded. It can be seen that queries regarding Modules, scholarships, and admission (CAO, advanced entry requirements, etc) tend to be asked more than three times per week. While intents such as accreditation, Erasmus/exchange programs, and job prospects are asked less frequently.

From Fig 3.3, it can be seen that different questions take different times to answer. Stakeholders mentioned that queries on the topics like stream/discipline, admissions, scholarships, and modules take in excess of 5 minutes to answer. Topics like jobs, accreditation, internships, and Erasmus/exchange programs can be answered within 3 minutes or lesser.

During graphical analysis, it's important to know that the chatbot can be accessed by students all around the world at the same time. Once developed, the chatbot would have the capability to address the queries of different students from a single place, all within the Trinity website.

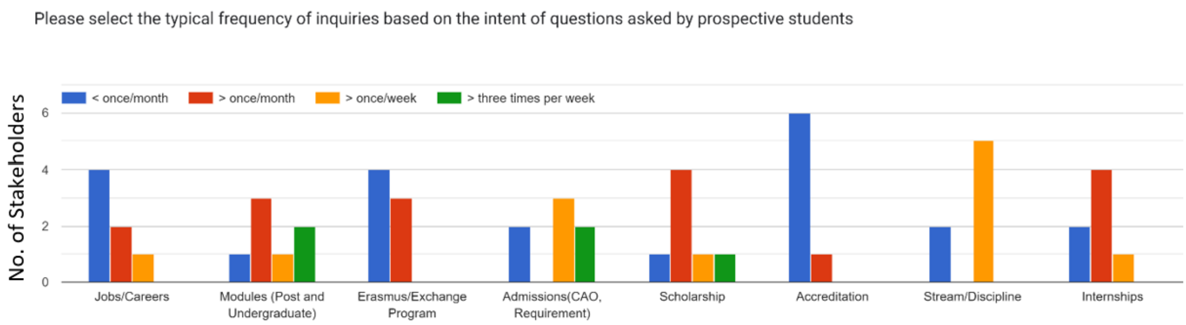

Figure 3.2: Frequency of queries asked based on the intent of the questions

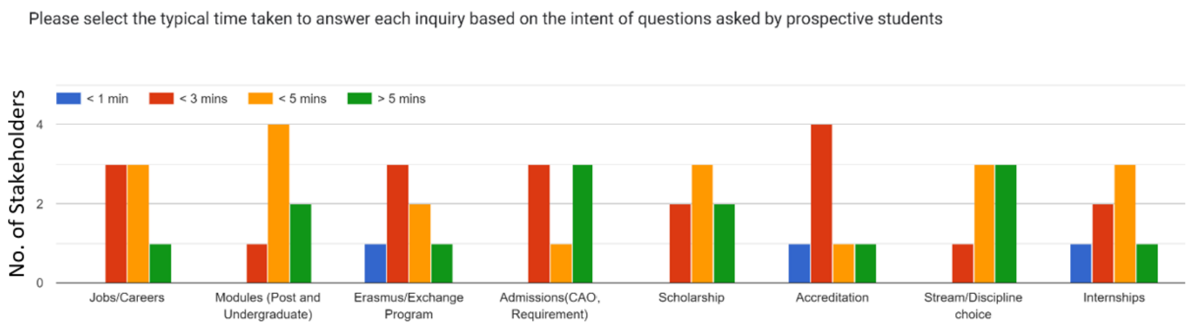

Figure 3.3: Time taken to answer each query based on the intent of the question

Therefore, it is beneficial to calculate the data from stakeholder research cumulatively to capture the aggregate effect. In this project, "the total number of queries received" and "the total time taken for 7 stakeholders to answer" values are recorded and calculated. By scaling up this approach, the overall impact can be calculated for all key staff the academic professors and school staff at the School of Engineering.

A metric is to be calculated to give a rough estimation of how much time is spent by the survey stakeholders in answering queries, therefore table 3.1 was constructed. The formula to calculate the total time spent in hours for the 7 stakeholders is given in equation 3.1. A few assumptions have been made to estimate more accurately.

- *Query Frequency per Month:* Evaluating the graphs provided in Fig 3.2, each column on the bar chart represents a corresponding frequency option. Assuming that a month has 4 weeks, stakeholders who selected the option ">once/week" could have received any frequency between 4 and 12 queries a month. The least possible values have been selected for each frequency option in order to establish a standard while calculating and evaluating the impact on the least busy month. Consequently, the following frequency calculation values per month are considered: **"<once/month"** $= 1$, **">once/month"** $= 2$, **">once/week"** $= 4$, and **">thrice/week"** $= 12$.
- *Time spent per query:* An important presumption in this section is that average query time rather than individual query time is being taken into account. It is crucial to keep

in mind that although one person can need 10 minutes to answer a question, another person from the school of engineering might only need 2. Therefore, a more generalized approach for the estimation is incorporated and the **mean of the time/query** is calculated for the 7 participants. The graph 3.3 illustrates that stakeholders were provided with a selection of values to choose from, similar to a frequency graph. However, the option ">5 minutes" had a broad range of potential values. From the stakeholder interviews, the School of Engineering office stated that responding to complex queries takes an average of 10 minutes. As a result, the following time limits have been selected for each option: **"<1 min"** = 1 minute, **"<3 mins"** = 3 minutes, **"<5 min"** = 5 minutes, and **">10 mins"** = 10 minutes.

$$Total\ Time(hr)\ for\ 7\ stakeholders = \frac{Query\ Frequency}{month} * \frac{Time/query}{7} * \frac{1}{60} \tag{3.1}$$

These assumptions have been incorporated in the equation 3.1 and the table 3.1 is constructed. As mentioned, the cumulative data is analyzed and it can be seen that taking the minimum conditions results in 164 total queries amongst 7 stakeholders (4 technical department and 3 school officers). Using the time taken to answer each query, and the frequency, the total time taken in hours comes out to be 15.8 hours per month for the 7 participants or **2.25 hours per person**.

Table 3.1: Estimation on the time spent on answering queries by the 7 stakeholders

Intent	Frequency/month	Time/query(min)	Total Time(hr)
Jobs/Careers	12	4.9	1.0
Modules & Courses	35	6.1	3.6
Erasmus/Exchange	10	4.3	0.7
Admissions	38	6.3	4.0
Scholarship	25	5.9	2.4
Accreditation	8	4.0	0.5
Stream/Discipline	22	6.9	2.5
Internship	14	4.6	1.1
Total	**164**	—	**15.8**

On the Trinity Engineering website, the breakdown of academic professors and school executive officers is given in table 3.2. The academic professors are inclusive of assistant professors, associate professors and professors. From the table, it can be seen that the total key staff that students tend to contact directly regarding queries is 75 staff members (65 technical professors + 10 departmental executive officers). Additionally, there are 9 school officers from the main office of Trinity Engineering which raises the total number to **84 key staff members**. If we consider the entire School of Engineering, the total time spent can add up to **189 work hours per month**. This is quite a significant number per month, considering the lower limit conditions that have been chosen. The number of admission queries tends to increase closer to the months of spring or autumn intake, therefore the time spent will increase in these months. A key note is that, this is generalized data and that responses can vary on an individual basis. The large dataset of all the key staff members was considered as it can offset the variance and noise from individual data.

Table 3.2: Academic Professors and Executive Officers breakdown in Trinity Engineering

Department	Academic Professors	Executive Officers
Civil Structural & Environmental	23	3
Mechanical, Manuf & Biomedical	28	4
Electronic & Electrical	14	3
Total	**65**	**10**

3.3 SWOT Analysis

Considering the stakeholder research opinions and quantitative results, a SWOT analysis can be constructed for the AI Chatbot.

- *Strengths:* The chatbot has no fixed timings per se and can run 24/7 on the website as long as there are functioning servers. This can be quite beneficial to high-school graduates who are looking for quick answers. There is an argument that can be made from the stakeholder research, that the AI Chatbot can reduce the overall time, the key stakeholders spend on answering admission queries. Any savings in time can directly result in overall cost savings and efficiency. [3] Moreover, the chatbot can alleviate the redundancy of students inquiring about basic information by directing them to the exact resources available on the Trinity website. This allows for professors to focus on other priorities in relation to academia. If successfully developed and scaled, it can answer a large number of queries simultaneously from different students as compared to individual responses in place in the current system. Incorporating rich text messaging and an intuitive GUI can deliver personal responses to the end users.

- *Weakness:* A major concern from the stakeholder analysis is the lack of emotional intelligence and human parity that the Chatbot may possess. There are possibilities that complex questions may not have satisfactory responses, which can lead to frustration for the users. There is a high chance of encountering technical difficulties when developing and integrating the Chatbot.

- *Opportunities:* There is an opportunity to improve the overall student experience in the admission process, which can be quite arduous and overwhelming. With the rapid development and Artificial Intelligence, Trinity College Dublin can set a standard by innovating and exploring AI Chatbot technology and reap its benefits.

- *Threats:* There are security and privacy concerns as users worry that the sensitive data entered could be mishandled, in violation of the General Data Protection Regulation (GDPR). A financial threat is also plausible as the additional cost of implementation can increase over the software development lifecycle (SDLC).

Upon analyzing the feasibility study (SWOT analysis) and the requirements from the stakeholder research, it is established that the AI chatbot technology can be explored. There is a need to develop an agent to aid prospective students in access to information. Along with precision and accuracy, the AI conversational agent should be enhanced to better engage with and captivate the attention of the students with appealing GUI.

4 Technical Development

The technical development for this project revolves around the Google Dialogflow construction and its knowledge base curator. This consists of the pre-built intents, entities, and the various responses that the agent delivers to the end-user.

4.1 Development of Google Dialogflow

The development of the AI conversational agent in Google Dialogflow was based on the prospective student's section on the Engineering website sitemap. Stakeholder opinions and results influenced the development of the AI Chatbot architecture and GUI features.

4.1.1 Intent Development

The intents were constructed by mapping the sitemap of the School of Engineering and its corresponding headings, as shown in Fig 4.1.

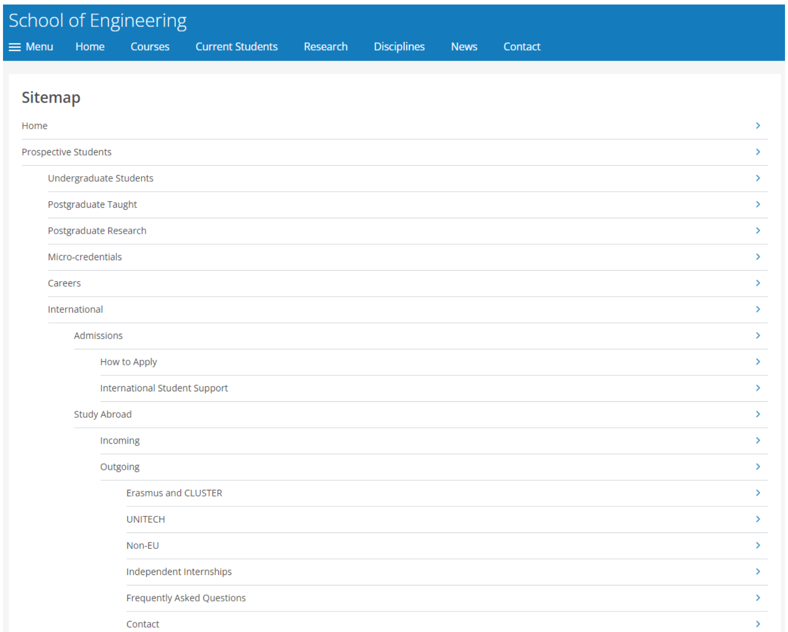

Figure 4.1: Sitemap of the Trinity Engineering Website [45]

A total of 32 intents were incorporated with 15 training phrases for each intent to cover a large range of queries that can be asked. some intents have been integrated after development can be found in the appendix A1.1. The red label given to the welcome and goodbye intent indicates the priority intent matching. The priority ranges from red to blue (high to low). For TrinityBot, the intent matching confidence score threshold is set at 0.4. If the confidence score is >0.4, the appropriate intent is matched but if <0.4, the FAQ section is searched and further no results in triggered the default intent.

The table in 4.1 shows us all the 32 intents integrated with TrinityBot. The purpose of each intent can be derived from its name. For example, "Prospective CAO", has information targeted at students with inquiries regarding Central Applications Office (CAO).

Table 4.1: List of 32 intents created and developed in TrinityBot

No.	Intent	No.	Intent
1	Default Fallback	17	Postgraduate Diplomas Courses
2	Default Welcome	18	Postgraduate PHD
3	Goodbye	19	Postgraduate Scholarship
4	Prospective CAO	20	Postgraduate Taught Courses
5	Prospective Careers	21	Postgraduate Taught Biomed
6	Prospective Contracts	22	Postgraduate Taught Computational
7	Prospective E3	23	Postgraduate Taught Environmental
8	Prospective English	24	Postgraduate Taught Mechanical
9	Incoming Erasmus	25	Postgraduate Taught Structural
10	Incoming Handbook	26	Postgraduate Taught Sustainable
11	Incoming Module Contacts	27	Postgraduate Taught Transport
12	Incoming Non-EU	28	Prospective Student Admission
13	Prospective Incoming Students	29	Prospective Testimonials
14	Prospective Internships	30	Prospective Undergraduate
15	Prospective Micro-credentials	31	Undergraduate Courses
16	Prospective Postgraduate	32	Undergraduate Scholarship

4.1.2 Conversational flow of the AI Chatbot

The design principles for the conversational flow are based on the stakeholder research conducted and the intents of the website's sitemap. The conversational flow is designed with a key focus on reducing technical errors. An error can be deemed technical if the chatbot does not give the users the appropriate responses. For example, a technical error is if a prospective **undergraduate** student posing a query about courses, receives a response regarding **post-graduate** courses.

To avoid these errors, the conversational agent flow is designed with care and is shown in Fig 4.2. Initially, the welcome intent is displayed and prompts the user to choose between undergraduate and postgraduate students. This is to filter the end user's queries according to their interests through the help of the events, input and output context features. This assists the AI chatbot in responding to individual queries in accordance with the status of the students. It also avoids errors when the users ask about a common topic that can relate to both undergraduate and postgraduate that is an added advantage. For example, information regarding scholarships is to be catered differently based on the student's education level.

Subsequently, the user is placed in either the prospective undergraduate or postgraduate block by the events "Prospective_Undergraduate" or "Prospective_Postgraduate" respectively. The lifespan for each of these blocks is set to 10 in TrinityBot which allows for follow-up questions in the same block. If the undergraduate option is clicked, the user is put in the "Prospective_Undergraduate" event. In reference to Figure 4.2, undergraduate prospective users can ask queries in relation to Admissions, Courses, Erasmus and Non-EU Exchange programs, career prospects, fees, scholarships, finances, contact information, etc for undergraduate studies.

However, if the postgraduate option is clicked, the "Prospective_Postgraduate" event is triggered which sends the user to the "Prospective Postgraduate" block. In this block, the user can ask the AI chatbot for information regarding Taught postgraduate courses, Ph.D. research opportunities, short-term micro-credentials, admission inquiries, Career prospects, scholarships, finances, contact information, etc in postgraduate studies.

If no intent is matched, AI Chatbot searches through the FAQ section to give a response that can answer the query. Then, if there is no match in the FAQ section, the Chatbot triggers the default intent, mentioning that the asked query could be outside the scope of the Chatbot.

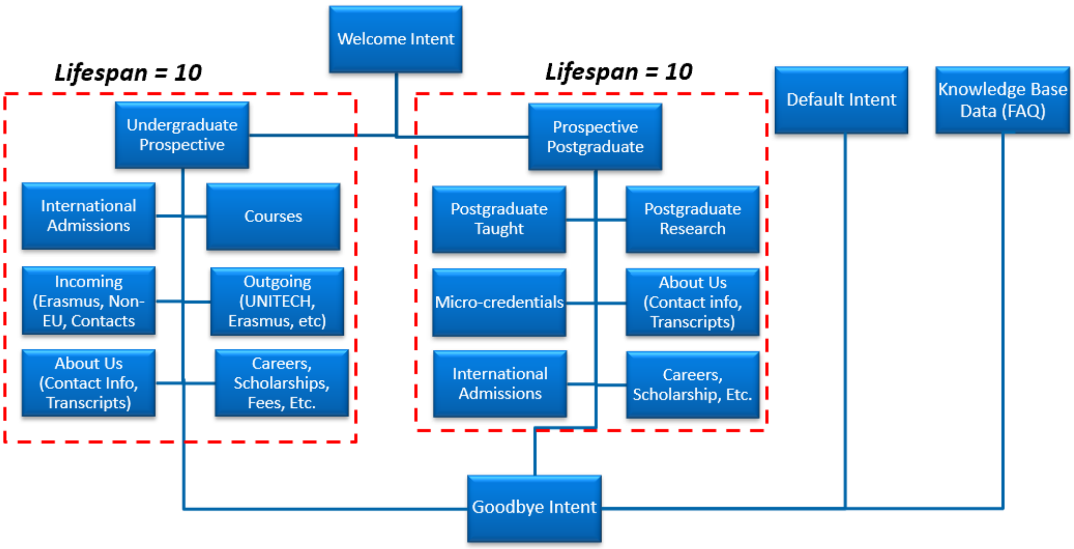

Figure 4.2: Developed Conversational flow of the AI Chatbot

4.1.3 Entity Development

In TrinityBot, there are in total 5 entities to help the chatbot extract parameters and assist in intent-matching. These 5 entities can be seen in the table 4.2. Furthermore, an example of the list of Taught Postgraduate courses at the School of Engineering, Trinity can be seen in A1.2

Table 4.2: List of Entities formed in TrinityBot

No.	Entity
1	Diplomas Courses
2	Micro-Credentials
3	Postgraduate PHD Research
4	Taught Postgraduate
5	Undergraduate Streams

4.1.4 Response Development

The response section can be created in two formats as mentioned in the literature review. For TrinityBot, a wide range of custom payloads has been programmed to improve the GUI, resulting in a better user experience. In Fig 4.3, an example has been shown in which the end user queries where the information regarding the Erasmus Program can be found as an incoming undergraduate student. The AI chatbot responds accordingly in a very intuitive manner.

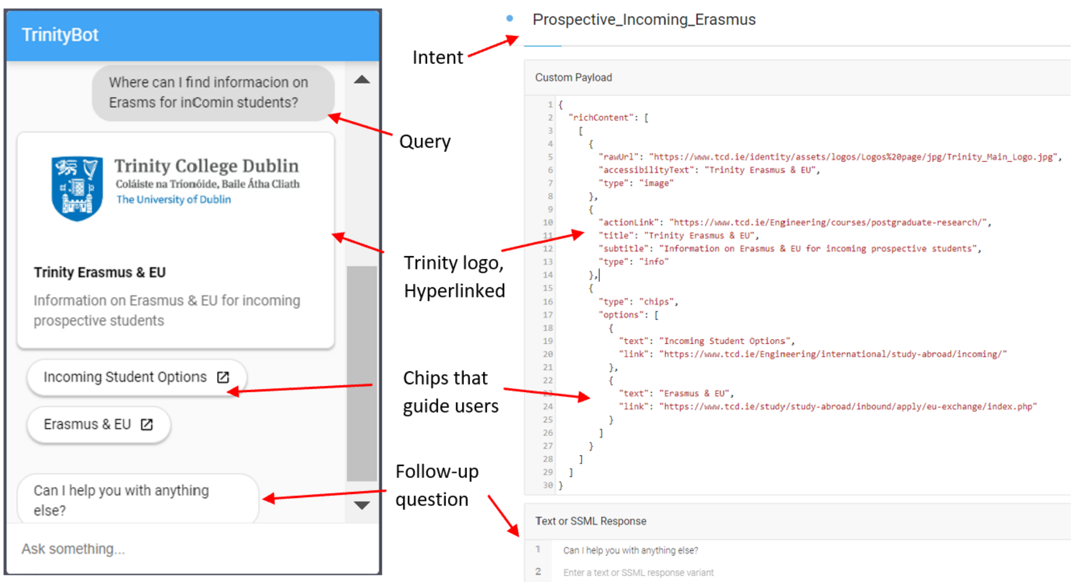

Figure 4.3: Coded Custom Response for the "Prospective Incoming Erasmus" Intent

It can also be seen that the user has misspelled the word "Information" and "Erasmus", while also using a capital "C" in the word "Incoming". The AI Chatbot, with its strong NLU, has the capability to ignore grammatical errors and understand what the end user is asking, and match the intent accordingly. The chatbot features and the way it has been coded in Dialogflow as a custom is also shown in Fig4.3. The utilization of chips allows the developer to hyperlink different web pages to the chatbot, to redirect the users to the relevant information. Since the lifespan has been set to 10 as mentioned in the technical architecture question, a follow-up question like "Can I help you with anything else" is set. Using input and output contexts, the follow-up question can be rerouted to the undergraduate block for another chatbot session. The TrinityBot also has a blue banner and a message-send button that brings a positive experience for the students. Users can also scroll up to see past queries and responses.

4.2 FAQ Section Development

The FAQ section of the Chatbot was taken from the Python Data Curator. The integration was done through a simple ".CSV" file and the priority of matching was set accordingly. In TrinityBot, the FAQ section matching confidence threshold was set at 0.2 as shown in Fig A1.6 Therefore, if a query has a confidence score of <0.4 but >0.2, then the FAQ section matching is triggered to find a response. However, if FAQ matching fails, then the default intent is triggered stating that the query is out of the scope of TrinityBot and that the user can contact the school officers directly via email. Appendix Fig A1.3 shows a small sample of the FAQ section. In total 35 questions and their answers were entered. The functionality allows for updating the FAQ knowledge base. In relation to TrinityBot, the school office can update the knowledge base of the AI chatbot to have the latest relevant information. Figure 4.4 shows an example of how a query matched with the FAQ section.

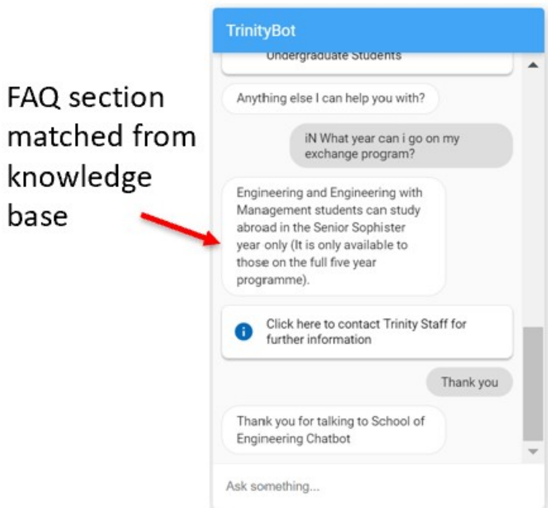

Figure 4.4: Erasmus Query that matched with the FAQ section

4.3 Integration and Embedding

From the literature review, Google Messenger was selected to integrate the AI conversational agent into the Trinity Engineering website. After the development of the AI conversational agent, the integration requires the developer to insert a Javascript file that contains the script source and other basic details of the chatbot. Fig A1.8 shows the messenger code that was inserted into the PHP server in the School of Engineering at Trinity. The embedding of the AI Chatbot to the Trinity servers was conducted primarily to see if the security systems in place at Trinity College Dublin would stop the third-party conversational agent from working as intended. The insertion of the Chatbot to the server made the agent public, however, it was made hard to access by users as it required the exact URL. In this case, the URL is "`https://www.tcd.ie/mecheng/test/`", which allowed the select stakeholders access to the conversational agent to try and test it out.

Fig 4.5 shows the successful and effective implementation of the AI chatbot on the Department of Mechanical, Manufacturing & Biomedical Engineering website. Here, the chatbot is queried regarding the available scholarship to postgraduate students and it responds as intended. The user can close the AI chatbot in the bottom right of the screen if they wish to continue using the website normally. Further pictures, like the initial screen that the chatbot displays and querying the undergraduate block can be found in the appendix A1.9.

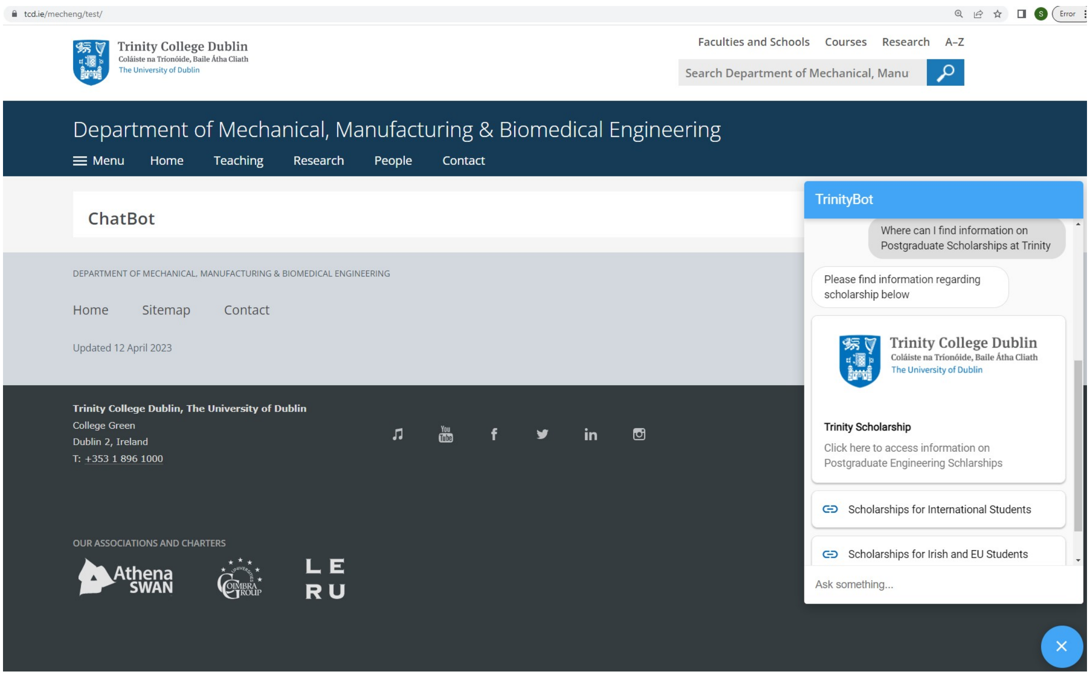

Figure 4.5: Embedding AI Chatbot in the Trinity Engineering Website

5 Results

5.1 Accuracy Metrics

The accuracy metrics for TrinityBot are divided into two parts. The intent matching and FAQ knowledge base. Since both these methodologies are integrated, they are tested separately and two modified confusion matrices are constructed.

5.1.1 Intent Matching Methodology

For TrinityBot intent-matching, 3-Fold cross-validation has been conducted since it has been tested with 32 intents and 15 training phrases. [46] In this technique, the training dataset was divided into 3 segments of the same sizes for 3 iterations. 2 of these segments were used for training (10 training phrases), and 1 for validation or testing (5 testing phrases). In total, there were *32 intents* *5 *testing phrases* *3 *iterations* = 480 testing queries entered in the Trinity Chatbot. The cross-validation was performed manually and recorded in Microsoft Excel. A keynote is that no additional validation of the AI Chatbot was conducted during testing. The functionalities such as buttons and hyperlinks were initially tested before recording the results to see if the agent worked as intended. The recorded data used to calculate the precision, recall and F1-score can be found in the appendix in Fig A1.10.

Table 5.1: F-1 Score of Intent-Matching methodology in TrinityBot

Iteration No.	Precision	Recall	F1-Score
1	0.910	0.820	0.865
2	0.900	0.840	0.870
3	0.890	0.820	0.855
Mean	**0.900**	**0.827**	**0.864**

From table 5.1, it can be seen that the mean F1-Score for this methodology from the 3 iterations of testing was **0.864**. When comparing the results to Fig 2.2, the bench-marked accuracy of TrinityBot is in accordance with the literature review. Moreover, the precision of

0.9 was higher than the recall of 0.827. This in turn means that Trinitybot's intent-matching capabilities can accurately identify and respond to most of the user's queries. However, there is a possibility that it may not be able to identify some pertinent intents and may miss relevant intents when answering.

5.1.2 FAQ Knowledge-Base

For testing the FAQ section of Dialogflow, a more traditional list of questions and answers was asked to the Chatbot. 5 queries resembling each of the 35 pre-built questions was asked to the agent over 3 iterations. This resulted in querying of 35 q*pre-built questions**3 *iterations* *5 *testing queries* = 525 testing queries. With the inclusion of the default fallback intent, a modified confusion matrix is constructed. The results are shown in table 5.2. The mean F-1 score for the FAQ section is 0.874, which is in accordance with the literature review of Google Dialogflow in Fig 2.2. The precision and recall metrics for this task are both satisfactory, with scores of 0.878 and 0.87 out of 1, respectively. The tabular format of the confusion matrix can be seen in the appendix A1.11

Table 5.2: F-1 Score of Intent-Matching methodology in TrinityBot

Iteration No.	Precision	Recall	F1-Score
1	0.870	0.860	0.865
2	0.890	0.860	0.875
3	0.875	0.890	0.883
Mean	**0.878**	**0.87**	**0.874**

When comparing the mean F-1 score of the intent-matching methodology (0.864) to the FAQ knowledge base (0.874), it can be seen that the FAQ section has better accuracy or a higher F1 score. Therefore the combination of intent-matching and FAQ knowledge base gives great accuracy in finding the relevant information when the student queries TrinityBot.

5.2 Error Analysis

For the bench-marking of the chatbot, there were key considerations when performing the testing. These considerations are primarily:

- *Data logging Errors*: Since most of the data logged is manual, there is a possibility of incorrect data entry while querying or data entry into Excel. Numerous testing iterations have been carried out to avoid this issue with the conversational agent.
- *Technical Errors*: Pretesting of the chatbot to ensure that the conversational flow is as intended was done before actual testing. This avoided any issues regarding the flow of

the chatbot when testing and kept the queries within the scope of the project.

5.3 Future Improvements and Developments

There are some future improvements that can be implemented to this AI Chatbot technology. This project emphasized the feasibility, validity of the technology and successful implementation of this agent on the Trinity Engineering website. With most of the aspects covered and achieved, there are potential developments and future work.

- Enhance natural language processing: The natural language processing of TrinityBot can definitely be improved with more optimized parameters and a refined conversational flow. This will allow the AI Chatbot to answer more accurately and achieve a higher F1 score.

- More personalized user experience: The addition of more rich text messages, intuitive graphics and features like cards or carousels to provide a visually appealing and engaging experience for your users.

- Expanding Chatbots Functionalities and Scope: TrinityBot can be expanded from just the School of Engineering to implement a more generalized AI Chatbot that is hosted on the main Trinity website and server. Additional functionalities such as the collection of prospective students' email IDs can be integrated with the conversational agent to contact the user with more personalized emails. This would in return have a far greater outreach to all prospective students, resulting in benefits to stakeholders in different departments (School Officers, Students and Professors) at Trinity College Dublin.

6 Conclusion

In conclusion, TrinityBot has achieved its project objectives with satisfactory performance and has proven to be an effective tool for addressing queries from prospective high school and graduate students. Its practicality was established through the comprehensive feasibility study and its embedding on Trinity servers, which demonstrated its scalability and potential for deployment on different platforms like Facebook, Slack, etc. With an F1 score **0.864** and **0.874** for intent matching and FAQ knowledge base respectively, the chatbot responds to the student queries with high confidence, accuracy and relevancy. Given that key staff members spend an average of **190 cumulative work hours per month** answering queries, TrinityBot has the potential to be highly advantageous in saving valuable academic time by streamlining the process of addressing basic questions and reducing redundancy. The stakeholders' concerns about the chatbot being fragile were addressed by implementing a well-defined conversational flow, resulting in accurate responses to a wide range of queries. The user-friendly interface and intuitive graphical features have enhanced the student experience and improved the School of Engineering's recruitment efforts.

With the rapid evolution of AI technology, chatbots are predicted to play a crucial role in the education sector by providing personalized support to students and in turn smoothening the exhaustive admission processes. With the successful integration of Trinitybot and subsequent scaling of the technology, Trinity College Dublin can set a new standard and precedence in the Irish Tertiary Education sector for CRM systems. Trinity can expand and provide access to innovative solutions that enhance the student experience and furthermore attract new students to the School of Engineering by embracing and incorporating advanced AI technologies while automating the existing systems.

Bibliography

[1] L. Xu, L. Sanders, K. Li, and J. C. L. Chow, "Chatbot for health care and oncology applications using artificial intelligence and machine learning: Systematic review," *JMIR Cancer*, vol. 7, no. 4, p. e27850, Nov. 2021. [Online]. Available: https://doi.org/10.2196/27850

[2] M. Barrett, L. Branson, S. Carter, F. DeLeon, J. Ellis, C. Gundlach, and D. Lee, "Using artificial intelligence to enhance educational opportunities and student services in higher education," *Inquiry: The Journal of the Virginia Community Colleges*, vol. 22, no. 1, p. 11, 2019.

[3] S. C. Man, O. Matei, T. Faragau, L. Andreica, and D. Daraba, "The innovative use of intelligent chatbot for sustainable health education admission process: Learnt lessons and good practices," *Applied Sciences*, vol. 13, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/4/2415

[4] A. Nurshatayeva, L. C. Page, C. C. White, and H. Gehlbach, "Are artificially intelligent conversational chatbots uniformly effective in reducing summer melt? evidence from a randomized controlled trial," *Research in higher education*, vol. 62, pp. 392–402, 2021.

[5] G. Dimitriadis, "Evolution in education: Chatbots," *HomVir*, vol. 3, no. 1, p. 47, Jun. 2020.

[6] S. V. Jin and S. Youn, "Why do consumers with social phobia prefer anthropomorphic customer service chatbots? evolutionary explanations of the moderating roles of social phobia," *Telematics and Informatics*, vol. 62, p. 101644, 2021.

[7] K. R. G. Raju, C. Adams, and R. Srinivas, "Cognitive virtual admissions counselor," *SMU Data Science Review*, vol. 1, no. 1, p. 6, 2018.

[8] D. Kaczorowska-Spychalska, "How chatbots influence marketing," *Management*, vol. 23, no. 1, pp. 251–270, 2019. [Online]. Available: https://doi.org/10.2478/manment-2019-0015

[9] J. Moor, *The Turing test: the elusive standard of artificial intelligence*. Springer Science & Business Media, 2003, vol. 30.

[10] A. Eleni and M. Lefteris, "An overview of chatbot technology," in *IFIP Advances in Information and Communication Technology*. Springer International Publishing, 2020, pp. 373–383. [Online]. Available: https://doi.org/10.1007/978-3-030-49186-4_31

[11] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, p. 100006, Dec. 2020. [Online]. Available: https://doi.org/10.1016/j.mlwa.2020.100006

[12] C. Bassett, "Apostasy in the temple of technology: ELIZA the more than mechanical therapist," in *Anti-computing*. Manchester, England: Manchester University Press, 2022. [Online]. Available: https://www.manchesteropenhive.com/view/9781526160720/9781526160720.00012.xml

[13] B. A. Shawar and E. Atwell, "Fostering language learner autonomy through adaptive conversation tutors," in *Proceedings of the The fourth Corpus Linguistics conference*, 2007.

[14] T. Xie, X. Yang, A. S. Lin, F. Wu, K. Hashimoto, J. Qu, Y. M. Kang, W. Yin, H. Wang, S. Yavuz, G. Wu, M. Jones, R. Socher, Y. Zhou, W. Liu, and C. Xiong, "Converse: A tree-based modular task-oriented dialogue system," 2022. [Online]. Available: https://arxiv.org/abs/2203.12187

[15] F. Masina, V. Orso, P. Pluchino, G. Dainese, S. Volpato, C. Nelini, D. Mapelli, A. Spagnolli, and L. Gamberini, "Investigating the accessibility of voice assistants with impaired users: Mixed methods study," *Journal of Medical Internet Research*, vol. 22, no. 9, p. e18431, Sep. 2020. [Online]. Available: https://doi.org/10.2196/18431

[16] S. K. Moore, "Is ai training outstripping moore's law?" Mar 2023. [Online]. Available: https://spectrum.ieee.org/ai-training-mlperf

[17] G. Bercaru, C.-O. Truică, C.-G. Chiru, and T. Rebedea, "Improving intent classification using unlabeled data from large corpora," *Mathematics*, vol. 11, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/3/769

[18] S. Perez-Soler, S. Juarez-Puerta, E. Guerra, and J. de Lara, "Choosing a chatbot development tool," *IEEE Software*, vol. 38, no. 4, pp. 94–103, Jul. 2021. [Online]. Available: https://doi.org/10.1109/ms.2020.3030198

[19] X. Liu, A. Eshghi, P. Swietojanski, and V. Rieser, "Benchmarking natural language understanding services for building conversational agents," 2019. [Online]. Available: https://arxiv.org/abs/1903.05566

[20] D. J. Patil, A. Shewale, E. Bhushan, A. Fernandes, and R. Khartadkar, "A voice based assistant using google dialogflow and machine learning," *International Journal of Scientific Research in Science and Technology*, pp. 06–17, May 2021. [Online]. Available: https://doi.org/10.32628/ijsrst218311

[21] N. Bhartiya, N. Jangid, S. Jannu, P. Shukla, and R. Chapaneri, "Artificial neural network based university chatbot system," in *2019 IEEE Bombay Section Signature Conference (IBSSC)*. IEEE, 2019, pp. 1–6.

[22] F. A. R. Finardi, G. M. De Holanda, C. Y. Adorni, M. V. P. De Nader, and K. R. Pinheiro, "Evaluating the application of a computer vision model in the customer service chatbot of an electric utility," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2021, pp. 1–5.

[23] A. Abdellatif, K. Badran, D. E. Costa, and E. Shihab, "A comparison of natural language understanding platforms for chatbots in software engineering," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 3087–3102, 2022.

[24] Łukasz Pawlik, M. Płaza, S. Deniziak, and E. Boksa, "A method for improving bot effectiveness by recognising implicit customer intent in contact centre conversations," *Speech Communication*, vol. 143, pp. 33–45, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639322000929

[25] "Quotas and limits | dialogflow | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Available: https://cloud.google.com/dialogflow/quotas

[26] K. V. Akhil, M. Shivakumar, and A. V. Reddy, "COGNITIVE NATURAL LANGUAGE PROCESSING ASSISTANT FOR POST-OPERATIVE CARE," *International Research Journal of Computer Science*, vol. 8, no. 6, pp. 125–131, Jun. 2021. [Online]. Available: https://doi.org/10.26562/irjcs.2021.v0806.001

[27] "Dialogflow es documentation | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Available: https://cloud.google.com/dialogflow/es/docs

[28] R. Vijayakumar, B. Bhuvaneshwari, S. Adith, and M. Deepika, "Ai based student bot for academic information system using machine learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 5, no. 2, pp. 590–596, 2019.

[29] "Training phrases | dialogflow es | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Available: https://cloud.google.com/dialogflow/es/docs/intents-training-phrases

[30] "Intent matching | dialogflow es | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Avail-

able: https://cloud.google.com/dialogflow/es/docs/intents-matching#:~:text=Dialogflow%20uses%20two%20algorithms%20to,is%20rarely%20a%20good%20option.

[31] S. A. Thorat and V. Jadhav, "A review on implementation issues of rule-based chatbot systems," *SSRN Electronic Journal*, 2020. [Online]. Available: https://doi.org/10.2139/ssrn.3567047

[32] G. Dialogflow, "Contexts | dialogflow es | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Available: https://cloud.google.com/dialogflow/es/docs/contexts-overview

[33] A. Sanjaya and H. Santoso, "Virtual assistant prototype for property sales," in *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*, 2021, pp. 334–338.

[34] "Agent validation | dialogflow es | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Available: https://cloud.google.com/dialogflow/es/docs/agents-validation

[35] C. Martin and Y. Li, "A scalable self-learning chatbot for classroom," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, pp. 0915–0921.

[36] "Integrations | dialogflow es | google cloud," 2021, [Accessed on 14 April 2023]. [Online]. Available: https://cloud.google.com/dialogflow/es/docs/integrations#:~:text=Dialogflow%20integrates%20with%20many%20popular,focus%20on%20building%20your%20agent.

[37] D. Peras, "Chatbot evaluation metrics," *Economic and Social Development: Book of Proceedings*, pp. 89–97, 2018.

[38] R. Regin, S. S. Rajest, T. Shynu *et al.*, "An automated conversation system using natural language processing (nlp) chatbot in python," *Central Asian Journal of Medical and Natural Science*, vol. 3, no. 4, pp. 314–336, 2022.

[39] M. I. Ismael, N. N. W. N. Hashim, N. S. M. Shah, and N. S. M. Munir, "Chatbot system for mental health in bahasa malaysia," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 2, no. 2, pp. 135–146, 2022.

[40] G. K. Reddy K, A. Pal, S. K. V, R. J, and S. K, "Cross domain answering faq chatbot," in *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, 2022, pp. 1–4.

[41] L. Fauzia, R. B. Hadiprakoso, and Girinoto, "Implementation of chatbot on university website using rasa framework," in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2021, pp. 373–378.

[42] V. V., J. B. Cooper, and R. L. J., "Algorithm inspection for chatbot performance evaluation," *Procedia Computer Science*, vol. 171, pp. 2267–2274, 2020, third International Conference on Computing and Network Communications (CoCoNet'19). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920312370

[43] S. Sossi Alaoui, Y. Farhaoui, and B. Aksasse, "Classification algorithms in data mining," *International Journal of Tomography and Simulation*, vol. 31, pp. 34–44, 08 2018.

[44] N. Lhasiw, N. Sanglerdsinlapachai, and T. Tanantong, "A bidirectional lstm model for classifying chatbot messages," in *2021 16th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2021, pp. 1–6.

[45] "Sitemap," 2020, [Accessed on 01 April 2023]. [Online]. Available: https://www.tcd.ie/engineering/sitemap/

[46] K. Kuksenok and A. Martyniv, "Evaluation and improvement of chatbot text classification data quality using plausible negative examples," 2019. [Online]. Available: https://arxiv.org/abs/1906.01910

A1 Appendix

A1.1 Intents, Entity and FAQ section

Sample Intents (Figure A1.1), **Taught Postgraduate Entity** (Figure A1.2), **Integrated FAQ section** ((Figure A1.3):

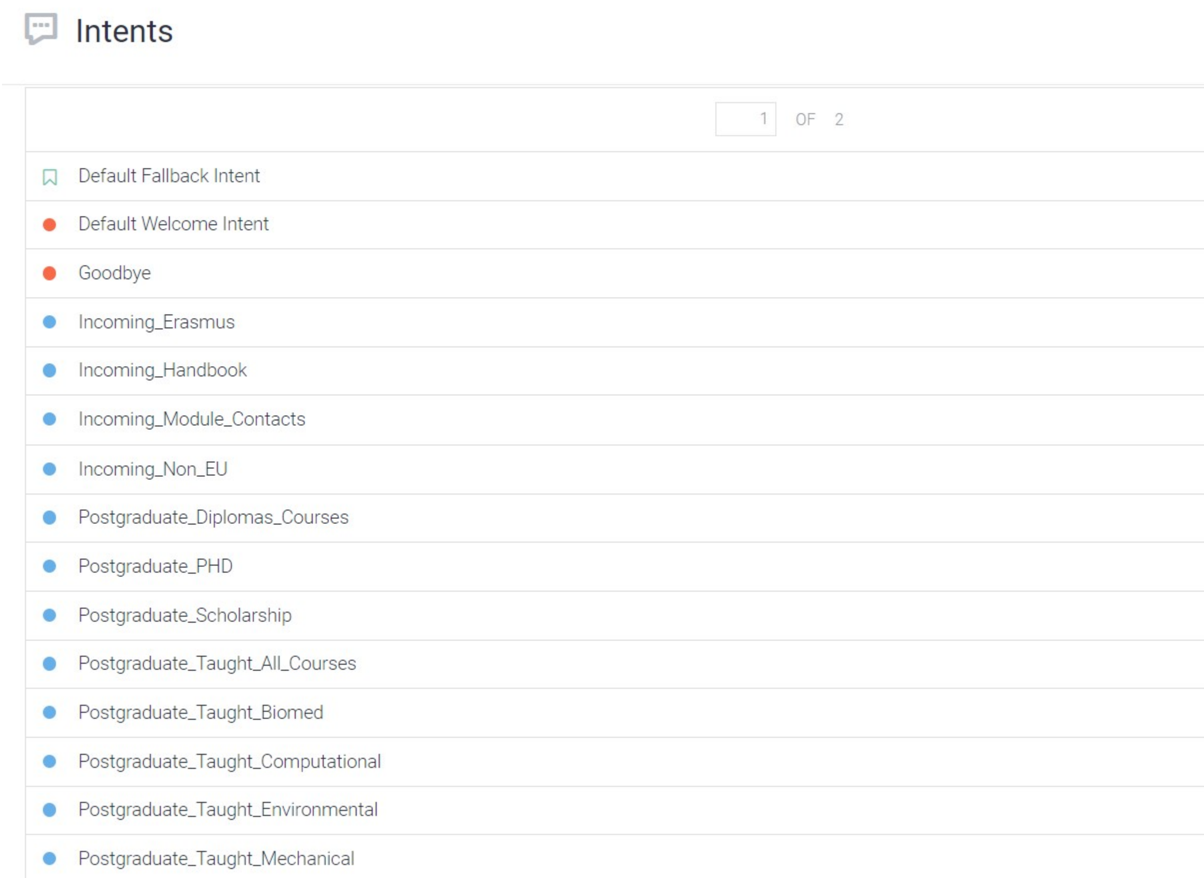

Figure A1.1: Sample of Intents in Dialogflow [45]

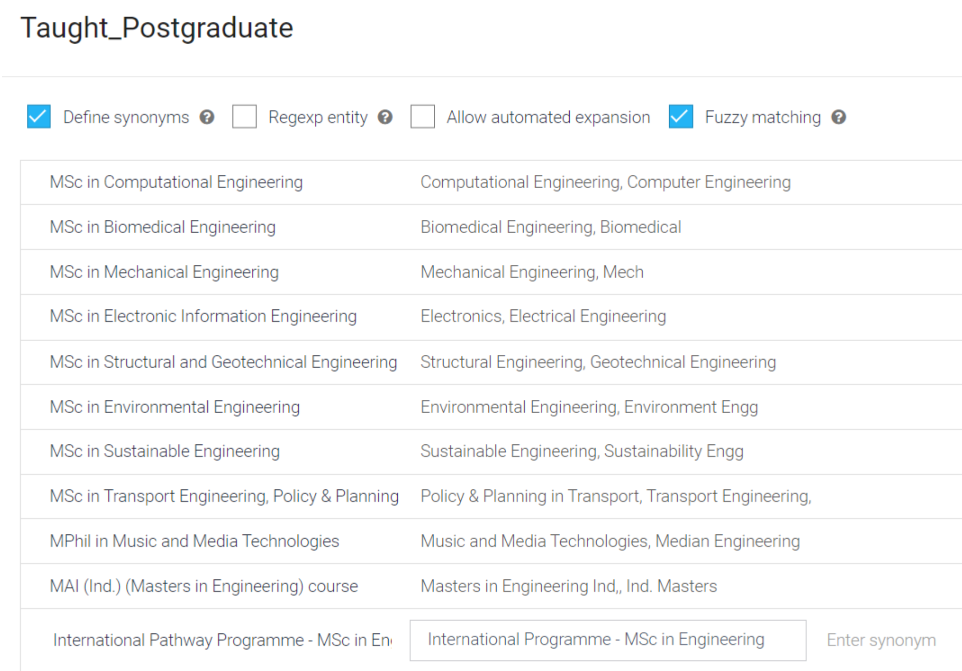

Figure A1.2: Taught Postgraduate Entity contents

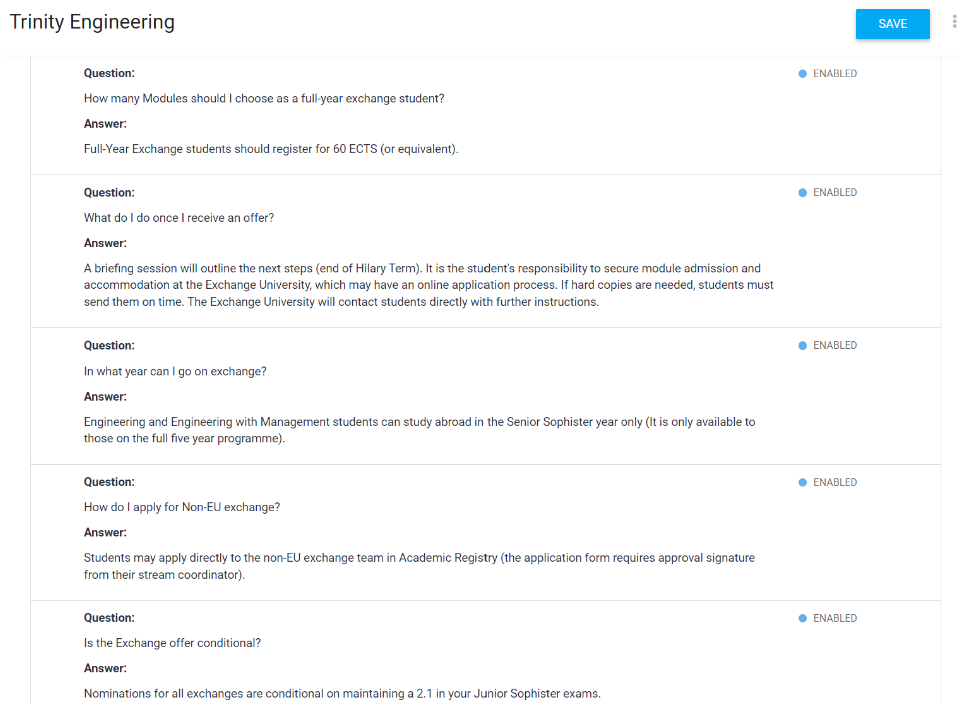

Figure A1.3: Integrated FAQ section

A1.2 Neural Network, NLTK and FAQ confidence score

Neural Network (Figure A1.4), **Natural Language Tool Kit** (Figure A1.5), **FAQ confidence score** (Figure A1.6):

```python
import torch
import torch.nn as nn


class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out
```

Figure A1.4: Model.py that represents the Neural Network model adapted from: [38]

```python
import numpy as np
import nltk
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

def tokenize(sentence):
    return nltk.word_tokenize(sentence)
def stem(word):
    return stemmer.stem(word.lower())
def bag_of_words(tokenized_sentence, words):
    sentence_words = [stem(word) for word in tokenized_sentence]
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1
    return bag
```

Figure A1.5: Natural Language Toolkit Python Script adapted from: [39]

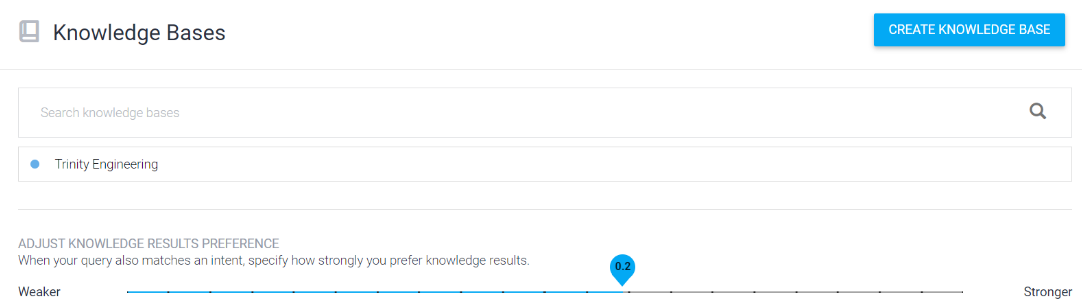

Figure A1.6: FAQ section confidence score

A1.3 Conversational flow, integration code and Implementation

Conversational flow example (Figure A1.7), **Integration Code of AI agent** (Figure A1.8), **Implementation of AI chatbot in Trinity Website** (Figure A1.9):

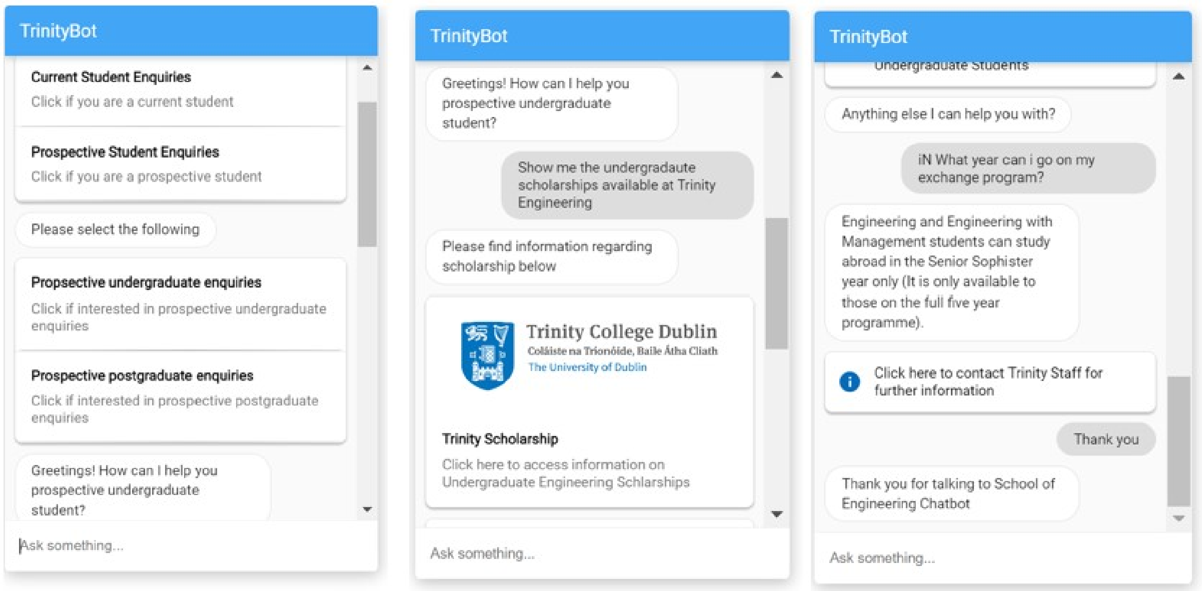

Figure A1.7: Conversational flow implemented on TrinityBot

```
<script src="https://www.gstatic.com/dialogflow-console/fast/messenger/bootstrap.js?v=1"></script>
<df-messenger
  intent="WELCOME"
  chat-title="TrinityBot"
  agent-id="64757a6c-a813-4a81-b906-40370061b08c"
  language-code="en"
></df-messenger>
```

Figure A1.8: Integration code to the PHP server at Trinity College Dublin

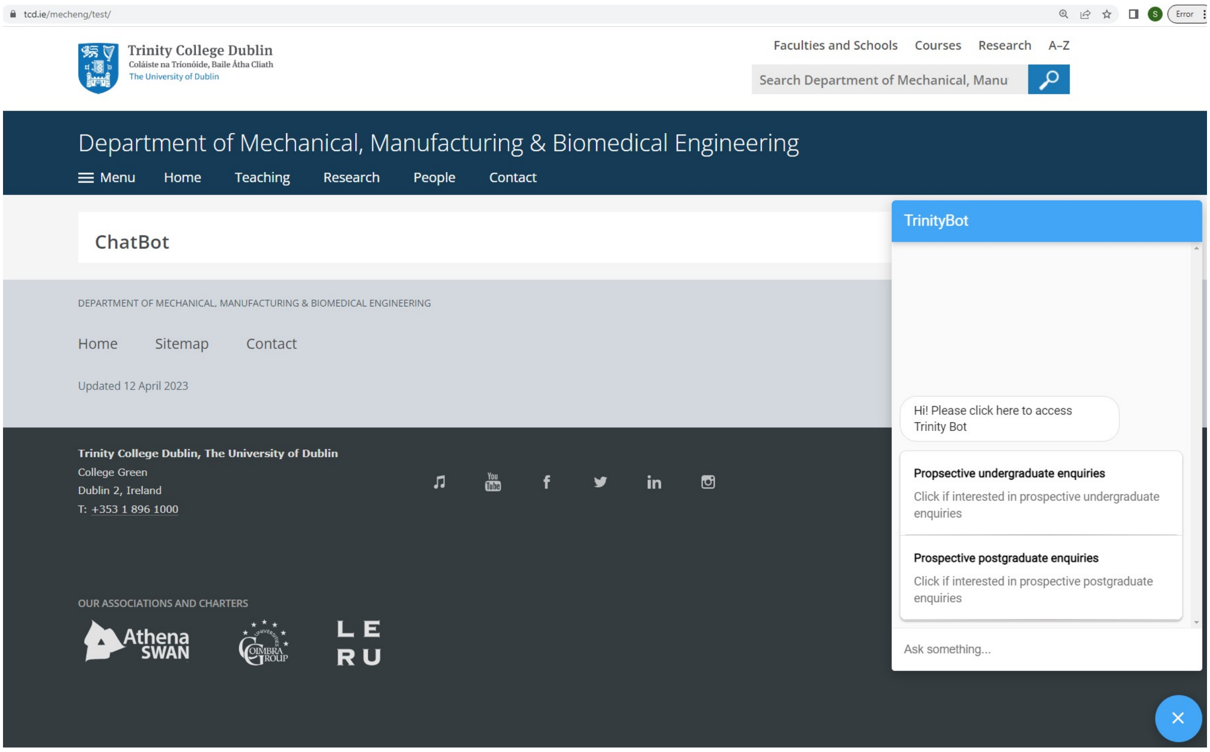

Figure A1.9: Implementation of AI Chatbot in the Trinity Engineering Website

A1.4 Modified Confusion Matrices

Modified Confusion Matrix for Iteration #2 in intent-matching (Figure A1.10), **Modified Confusion Matrix for FAQ section testing iteration #1** (Figure A1.11):

Intents (Iteration #2)	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)	Precision	Recall	F1-Score
Default Fallback Intent	5	0	0	0	1.00	1.00	1.00
Default Welcome	5	0	0	0	1.00	1.00	1.00
Goodbye	5	0	0	0	1.00	1.00	1.00
Incoming_Erasmus	4	1	0	0	0.80	1.00	0.90
Incoming_Handbook	4	1	0	0	0.80	1.00	0.90
Incoming_Module_Contacts	3	1	0	1	0.75	0.75	0.75
Incoming_Non_EU	4	0	0	1	1.00	0.80	0.90
Postgraduate_Diplomas_Courses	4	1	0	0	0.80	1.00	0.90
Postgraduate_PHD	3	1	0	1	0.75	0.75	0.75
Postgraduate_Scholarship	4	1	0	0	0.80	1.00	0.90
Postgraduate_Taught_Biomed	4	1	0	0	0.80	1.00	0.90
Postgraduate_Taught_Computational	4	0	0	1	1.00	0.80	0.90
Postgraduate_Taught_All_Courses	3	1	0	1	0.75	0.75	0.75
Postgraduate_Taught_Environmental	4	1	0	0	0.80	1.00	0.90
Postgraudate_Taught_Mechanical	5	0	0	0	1.00	1.00	1.00
Postgraduate_Taught_Structural	3	0	0	2	1.00	0.60	0.80
Postgraduate_Taught_Sustainable	5	0	0	0	1.00	1.00	1.00
Postgraduate_Taught_Transport	2	1	0	2	0.67	0.50	0.58
Prospective_CAO	5	0	0	0	1.00	1.00	1.00
Prospective_Careers	4	0	0	1	1.00	0.80	0.90
Prospective_Contacts	2	2	0	1	0.50	0.67	0.58
Prospective_E3	3	0	0	2	1.00	0.60	0.80
Prospective_English	4	0	0	1	1.00	0.80	0.90
Incoming_Students_Entry_Pathways	4	1	0	0	0.80	1.00	0.90
Prospective_Internships	3	1	0	1	0.75	0.75	0.75
Prospective_Micro-Credentials	4	0	0	1	1.00	0.80	0.90
Prospective_Postgraduate_block	5	0	0	0	1.00	1.00	1.00
Prospective_Student_Admission_Info	3	0	0	2	1.00	0.60	0.80
Prospective_Testimonials	4	0	0	1	1.00	0.80	0.90
Prospective_Undergraduate_block	3	0	0	2	1.00	0.60	0.80
Undergraduate_All_Courses	3	0	0	2	1.00	0.60	0.80
Undergraduate_Scholarship	4	0	0	1	1.00	0.80	0.90
Average					**0.90**	**0.84**	**0.87**

Figure A1.10: Recorded data for evaluation metric calculation for iteration #2

Question No. (Iteration#1)	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)	Precision	Recall	F1-Score
1	4	0	0	1	1.00	0.80	0.90
2	3	0	0	2	1.00	0.60	0.80
3	5	0	0	0	1.00	1.00	1.00
4	3	1	0	1	0.75	0.75	0.75
5	4	1	0	0	0.80	1.00	0.90
6	4	0	0	1	1.00	0.80	0.90
7	5	0	0	0	1.00	1.00	1.00
8	3	1	0	1	0.75	0.75	0.75
9	4	0	0	1	1.00	0.80	0.90
10	3	2	0	0	0.60	1.00	0.80
11	5	0	0	0	1.00	1.00	1.00
12	3	1	0	1	0.75	0.75	0.75
13	4	0	0	1	1.00	0.80	0.90
14	4	1	0	0	0.80	1.00	0.90
15	4	0	0	1	1.00	0.80	0.90
16	5	0	0	0	1.00	1.00	1.00
17	4	1	0	0	0.80	1.00	0.90
18	3	1	0	1	0.75	0.75	0.75
19	4	0	0	1	1.00	0.80	0.90
20	3	1	0	1	0.75	0.75	0.75
21	4	0	0	1	1.00	0.80	0.90
22	5	0	0	0	1.00	1.00	1.00
23	3	1	0	1	0.75	0.75	0.75
24	3	2	0	0	0.60	1.00	0.80
25	4	1	0	0	0.80	1.00	0.90
26	3	0	0	2	1.00	0.60	0.80
27	3	0	0	2	1.00	0.60	0.80
28	4	1	0	0	0.80	1.00	0.90
29	3	1	0	1	0.75	0.75	0.75
30	5	0	0	0	1.00	1.00	1.00
31	3	0	0	2	1.00	0.60	0.80
32	5	0	0	0	1.00	1.00	1.00
33	4	1	0	0	0.80	1.00	0.90
34	5	0	0	0	1.00	1.00	1.00
35	5	0	0	0	1.00	1.00	1.00
Average					**0.89**	**0.86**	**0.88**

Figure A1.11: FAQ Knowledge Base Benchmarking iteration #1