Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	**F21MP**
Type of assessment:	**Individual**
Coursework Title:	**User-Chatbot Interaction Analysis: A Comparative Study of Chatbot Experiences**
Student Name:	Jose Berlin Durai Yoseppu
Student ID Number:	H00409950

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(Jose Berlin Durai Yoseppu):* *Jose Berlin*

Date: *12/12/2023*

Heriot Watt University

F21MP

User-Chatbot Interaction Analysis: A Comparative Study of Chatbot Experiences

Author:

Jose Berlin Durai Yoseppu

Supervised by :

Ioannis Konstas

A thesis submitted in fulfilment of the requirements

for the degree of MSc. Artificial Intelligence

in the

School of Mathematical and Computer Science

December 2023

Abstract

Chatbots are computer programmes that use messages that is created to mimic and respond using human-like conversations with human users. Natural language processing (NLP) technologies are used to comprehend and respond to the user requests and queries (in a human-like way). NLP is a subfield of Artificial Intelligence (AI) which concentrates on natural language interactions, which acts as a bridge between computers and humans to understand what the computer program trying to say. Chatbots are now widely utilized in many areas, including customer service, retail, finance, and health care. They are targeted to automate routine tasks and give users quick and easy access to information, goods, and services. This project is about determining the user's preference in information delivery by implementing a chatbot for students which delivers the answers about "University course details". And also at the present moment, the students at university don't have any type of chatbots which are able to answer the questions about courses and module-related information efficiently. This research aims to investigate whether users prefer to receive Answers only, Sources only, or a combination of answers and sources from a chatbot. In order to evaluate this research firstly, we will plan and develop a chatbot in which students will be able to request and obtain information about different courses available at Heriot-Watt University in the above mentioned 3 different methods and we will be using human evaluation methods to evaluate our research.

Acknowledgements

My profound gratitude goes out to my supervisor, Prof. Ioannis Konstas, for his helpful advice, motivation, and support during the entire research process. I want to thank him for devoting many hours in helping me despite his intense workload.

Contents

Table of figures

1. Introduction

In the past few years, Chatbots have seen a substantial growth by the implementation of chatbots in wide range of purposes including information retrieval, customer service, and even in personal assistance. While these chatbots can deliver users with quick and easy access to information, they often struggle to generate meaningful and engaging conversations that fitting all the user's demands. One of the possible reasons is that many chatbots are not designed considering the user preferences for the type/kind of responses the users obtain from the chatbot. It remains unclear whether users prefer receiving only answers, only sources of the answers, or a combination of both (information and sources). The latter method, providing answers along with sources, may potentially enhance user trust in the chatbot's responses by allowing users to verify the accuracy of the information. But in order to know that for certain, we must have to understand how the chatbot responses affect users satisfaction, trust and their willingness to suggest the chatbot to others.

This research starts with developing a working chatbot which will be delivering information from Heriot-Watt university's course website in the three specified ways. And, to make this possible we will employ a chatbot which will be able to understand, process and generate answers for user queries related to Heriot watt courses.

As we can see in the Figure 1, the chatbot will allow users to ask questions and then the chatbot analyses the content/questions received from the user, then retrieve and deliver the information from the retrieved website data which has the possible answer in this research project, we aim to concentrate on this issue by examining the user preferences for chatbot responses. Specifically, this study intends to investigate the factors that influence users' preferences for various chatbot responses. To understand and determine how chatbots might be adjusted to deliver more interesting and beneficial interactions, we want to conduct a survey with the people who use chatbots. Here our chatbot will be using Question Answering technique to receive and deliver information to the users.

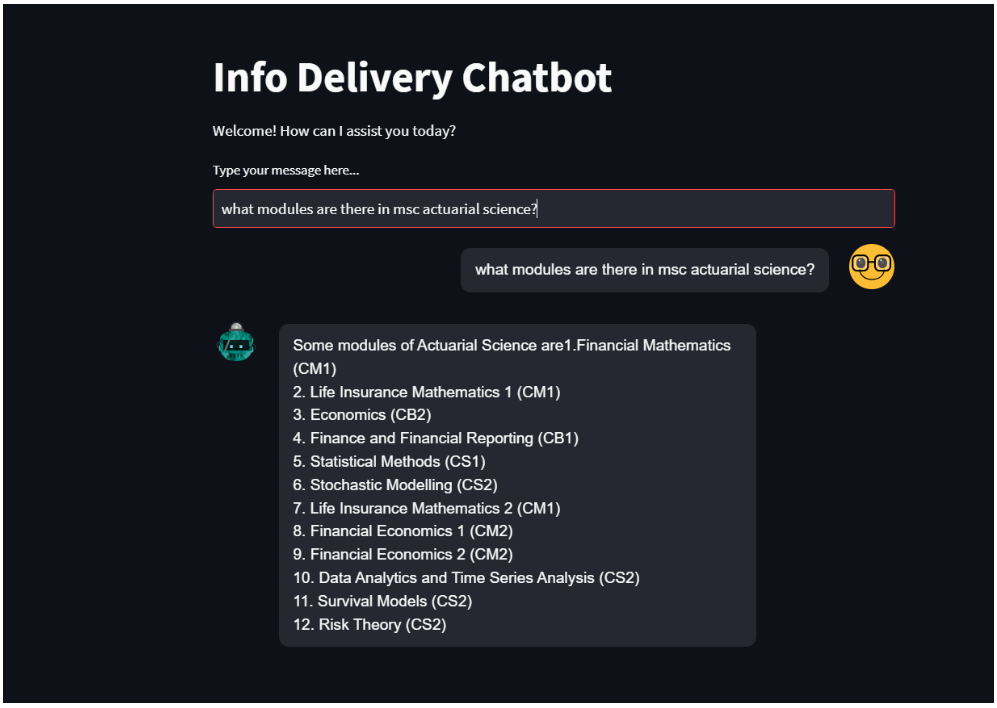


Figure 1: Example of our chatbot giving response as the user asked for the modules of MSc Actuarial Science course of Heriot Watt University

The key question here is whether users prefer to get only the answers they enquired for, or only the sources (links) which holds the enquiry's information, or both answers and their sources of the enquiry, from the chatbot. This study will address the lack of knowledge in user preferences for chatbot responses as well as provide knowledge for the design and creation of more user-friendly chatbots that can more effectively meet user expectations.

AIMS:

1. To investigate the preferences of users for various kinds of information delivery in chatbots.
2. To identify the limitations and strengths of answer only, sources only, and answers + sources chatbot
3. To identify the factors that affect user preferences and satisfaction with chatbot information delivery.

OBJECTIVES:

1. To build an NLP model that can derive and understand information from the Heriot Watt University course website.
2. To implement the model into chatbot that can deliver the information in 3 different methods i.e., answers only, sources only, and answers with sources.
3. To compare the reliability of the different types of responses through user testing.
4. To employ a user survey for analysing feedback from users on chatbot responds.
5. To determine areas where chatbot technology might be improved to better inform consumers.

This research project is significant because this research will be able to help in the advancement of building chatbots which are more useful, more able to satisfy user demands, and more gratifying overall.

2. Literature Review

2.1. Introduction:

Chatbots are computer software programmes that imitate the human communication with the help of artificial intelligence (AI) and machine learning (ML) algorithms. To advance this human communication imitating process these chatbots employ Natural Language Processing (NLP) techniques to grasp the meaning of the human language, analyse it and respond to user inquiries in a natural and insightful manner.

Here are some of the real-world applications of chatbots:

1. **Customer service**: Many organizations utilize chatbots to give rapid and efficient help to their clients. Here in customer service, these chatbots are designed to provide respond clients for the most generally asked questions in conversational and natural language.
2. **E-commerce**: Chatbots are also used in e-commerce to provide support customers in finding products, providing suggestions, and in completing orders.

3. **Healthcare**: Practitioners are also gradually increasing the utilisation of chatbots by using them to assist patients in scheduling appointments, delivering information on medical concerns, and examining patient health condition.
4. **Finance**: Chatbots are also being utilised by banking and financial organizations to provide support for consumers with account management, financial development plans, and also in investment counselling.

Overall, NLP chatbots provide a wide range of possible applications and can increase the efficiency and effectiveness of a variety of businesses. And when it comes to the Education sector, Chatbots can be used to provide assistance in course selection, student assistance, student/teacher mental health assistance and other administrative jobs. The purpose of the research is to investigate the user preferences in the types of responses supplied by an Chatbot.

2.2. Background of Chatbot:

A chatbot, often referred to as a conversational agent, is a component of computer software that can take a natural language input and produce a discussion in real time. Typically, a graphical user interface based on human computer interaction (HCI) concepts which is used for this human-chatbot connection. [1]

The first chatbot history goes even beyond the first personal computer was created, the first chatbot was launched by Joseph Weizenbaum. Joseph Weizenbaum built a chatbot in 1966 at the MIT Artificial Intelligence Laboratory, which was named as Eliza, a rule-based chatbot. Eliza pretended to be a therapist. Eliza was designed to recognise the keywords from the user's input and triggered the rules for output transformation. While creating chatbots, this specific approach of generating replies is still often employed. In an effort to replicate someone with paranoid schizophrenia, psychiatrist Kenneth Colby, who was then at Stanford University, wrote Parry chatbot after Eliza.[2]

Then A.L.I.C.E. was created by Richard Wallace in 1995 after drawing inspiration from Eliza, the chatbot was also known as Alicebot. A.L.I.C.E. was given the Loebner Prize [2.2.1] three times despite failing the Turing test [2.2.2], making it one of the most powerful programmes of its kind.

ActiveBuddy (a conversation-based interactive agents software company) created a chatbot called SmarterChild in the first decade of the twenty-first century. It was the first attempt to develop a chatbot that could offer users more valuable information, such as stock statistics, sports scores, movie quotations, and much more, in addition to fun. More than 30 million individuals used it, and it was made available through Windows Live Messenger and AOL (American Online). The ancestor of Apple's Siri and Samsung's S Voice is SmarterChild. The figure 2 shows the timeline of the chatbots.

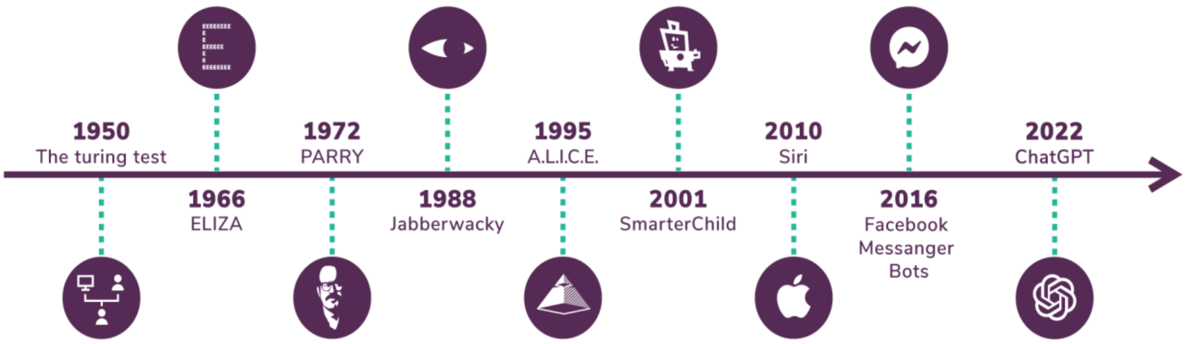


Figure 2: Timeline of most popular Chatbots since the year 1950 [3]

[2.2.1] The bot that scores the highest on the Turing Test is given the Loebner prize. The Loebner competition judges are having two conversations at the same time. One involves a human, while the other involves a bot. The participant who fools a judge the most proportion of the time wins the tournament.

[2.2.2] A Turing test is a method of evaluating the intelligence of a computer in which the sender (a person) is provided with responses from both a machine and another person (a recipient). During his time at the University of Manchester, Alan Turing developed the Turing test, which was published in his paper "Computing Machines and Intelligence" in 1950.[3]

The intelligent personal assistant Siri was developed by SRI International as a side project and eventually included by Apple into its iOS 5 for the iPhone. Together with Apple, tech giants Samsung and Google have created their own AI assistants, S Voice and Google Allo, respectively.

As we can see in figure 2, the current decade holds the most popular chatbot named ChatGPT (GPT-3), a question answering system which uses Deep Learning and ML algorithms, it is a Generative AI tool developed and released by OpenAI in November 2022. Even though its results are not 100 percent accurate, it surprised everyone with its remarkable ability to process and generate complete reply (responds) for the user requests, even for the complicated requests like pretending to be someone, ability to program, proving logics etc., It is considered as a great breakthrough in the history of Chatbot and AI (especially NLP). [4]

Similar to these above mentioned chatbots, we will be developing a chatbot but instead of open domain, our chatbot will be developed to handle conversations on a closed domain i.e., Heriot Watt course details and it will be using NLP techniques that will power the chatbot by providing ability to understand the users query and generate response from knowledge base i.e., processed data from website and answering queries of users using question answering method.

2.3. Related Studies:

Building systems that autonomously answers the questions asked by humans in natural language is the motivation of the computer science field in question answering (QA), which also includes the processing of information retrieval and natural language processing (NLP). A Question Answering system, typically a computer software, may generate its solutions by querying a database having structured data, typically a knowledge base. The majority of the time, question-answering algorithms can draw information from a group of unstructured natural language documents. These are achieved using ML and deep learning methods, which are based on neural networks, a subset of machine learning also referred to as artificial neural networks or synthetic neural networks.

The structure and the names are designed by trying to mimic the process of human brain, to be exact: imitating the communication of biological neurons between one another. Here in this section, we will be looking into similar studies related to ours i.e., Closed domain Question Answering chatbots.

The author Swathilakshmi, V., (2021) [5] suggested a closed domain question-and-answer system that primarily focuses on the educational domain. A question-and-answer system takes the user's queries, extracts the intents and entities from the input, looks up the responses in the database, and then presents the most relevant response. Here the database contains all queries relating to the subjects covered in the NCERT textbooks. Here in this study, the proposed QA system will first identify the subjects that the student is enrolled in before responding to their queries to gain more knowledge on the material that are covered in each individual's textbook. But here in our research, we will be creating a closed domain question answering chatbot which will be delivering information about the course details domain i.e., information about the Heriot Watt University's different course and its contents, instead of NCERT textbooks and also, we will be evaluating the chatbot using anonymous users unlike collecting any individual's information to increase the focus of the chatbot on particular subjects in this above discussed study.

The authors Tran, P., Nguyen, D., Tran, H.A., Nguyen, T. and Tran, T., (2023) [6] have proposed a system namely, "The Postgraduate Admission Advisory System of Ho Chi Minh City University of Food Industry, Vietnam". In the paper, two key models—IC model and MRC model—are used to create a Question Answering System (QAS) for the postgraduate entrance advisory system of a Vietnamese institution. Using Google Translate to translate from English to Vietnamese and carefully generating the HUFI-PostGrad corpus, the authors produced a sizable Vietnamese QA dataset. Basically, this system will get input from users and extract the keywords from the input and generate possible answers by looking into the database. Here the authors of this research, came to a conclusion that, in order to improve the quality of the QAS and postgraduate participation, it is essential to increase the amount and quality of the dataset. They do so by acknowledging the shortcomings of both IC and MRC methods. As in our chatbot we will be carefully pre-process the retrieved data and so that we can expect better accuracy from our chatbot.

The authors Zhou, X., Nurkowski, D., Menon, A., Akroyd, J., Mosbach, S. and Kraft, M., (2022) [7] have proposed a QA system which will be able to answer queries regarding contents of chemistry. In the development process of this QA chatbot they have implemented NER model to recognise entities from the user input and question classification model to categorize the user queries i.e., called intent classification. The article presents an extension of a question-answering system for chemistry that, when the knowledge graph is unable to provide information, automatically invokes the semantic agents that will handle the question with a default answer. Similar to this research, our chatbot will be having an NLG handler which will analyse the query, whether it comes under the domain or not. If the query doesn't come under the domain, then that query will be answered with a default answer.

The authors Satapathy, S.C., Raju, K.S., Shyamala, K., Krishna, D.R. and Favorskaya, M.N. (2019) [8]has implemented chatbot and experimented different answer retrieval methods including Question Matching techniques and Ranking Algorithm, SPRQL queries, HMM-based named entity recognizer in both open and closed domain chatbots. The author also noted that there has been almost equal research on closed and open domain QAS, with five articles concentrating on closed domain and the remaining four on open domain QAS. It was observed that because closed domain QAS is single domain in nature, it requires less processing to obtain correct results, making it more accurate. So that we can expect more satisfying results from our chatbot as it will be on a closed domain and also, we will be carefully pre-processing the info which I expect to lead to much higher accuracy in results.

Finally, the authors DERİCİ, C. A. N. E. R., AYDIN, Y. İ. Ğ. İ. T., YENİALACA, Ç. İ. Ğ. D. E. M., AYDIN, N. İ. H. A. L. Y. A. Ğ. M. U. R., KARTAL, G. Ü. N. İ. Z. İ., ÖZGÜR, A. R. Z.

U. C. A. N. and GÜNGÖR, T. U. N. G. A. (2018) [9] discusses the implementation of a Turkish QA system called HazrCevap, which aims to provide students with trustworthy and precise responses to their inquiries about the topics geography and biology. For comprehensive covering, the system combines a variety of online resources, such as Wikipedia and instructional websites, and incorporates multilingual resources. The system not only provides answers but also relevant information on the subject of the query and papers that are connected for further research. 65%–80% of papers holding responses were successfully retrieved by the system, according to experiments. 4,000 factoids and open-ended queries in the subjects of geography and biology were used to evaluate the system. The Google search engine was also provided the queries without any pre-processing in order to compare the outcomes with those of HazrCevap's system. HazrCevap's achievement rate was discovered to be about 10% higher than Google's as it was a closed domain system just like the author declared in [8].

The research discussed above show that question-answering systems/Chatbots have the ability to deliver precise and trustworthy information when it's implemented within a certain closed domain. They additionally bring focus to the systems' drawbacks, such as the requirement for high-quantity and high-quality datasets, requirement of precise pre-processing of the data, and the possibility of bias or insufficient knowledge base analysis. Numerous research also concentrates on certain languages or topic areas, which may restrict their application in other conditions. Overall, even though closed domain question answering systems have demonstrated promise in enhancing information access and assisting with educational and research work, more research is required to solve these issues and assure their usefulness across a larger range of topics and languages.

2.4. Evaluation Method:

2.4.1. Evaluation Metrics for this research:

Evaluating the chatbot system's efficiency and user satisfaction in providing information about university courses is essential to this research. In order to evaluate the effectiveness as well as the impact of the chatbot in a systematic way, the following measures have been determined:

1. **User Satisfaction Score:** Users' level of satisfaction with the chatbot's responds will be measured using a Likert scale in the evaluation questionnaire. It will take into account elements including information completeness, readability, and relevancy to the user's inquiry.
2. **Accuracy of Information:** This will assess the accuracy and relevance of the chatbot's responses to user queries. Accuracy can be measured based on user ratings in the questionnaire, indicating how well the chatbot's answers matched the users' information needs.
3. **User Preference:** Determining the preference for using the chatbot over traditional web browsing methods. This metric is crucial for understanding the chatbot's value proposition from the user's perspective.
4. **Trust in Responses:** Measured by users' ratings, this metric will indicate the extent to which users trust the information provided by the chatbot. Trust levels can influence the likelihood of users relying on the chatbot for future information needs.
5. **Recommendation Likelihood:** This will measure the likelihood of users recommending the chatbot to others. A high likelihood of recommendation can be an indicator of overall user satisfaction and the perceived usefulness of the chatbot.
6. **System Usability Scale (SUS):** A standard questionnaire used to evaluate the usability of the system. It provides a global view of subjective assessments of usability.

2.5. Chatbot Workflow:

As we can see in the figure 3, once the user sends the first input in Natural Language and the input goes to the Natural Language Understanding phase where the program will extract the intent and entities using the user's input, Then the input goes to the Dialogue Manager who manages the dialogue flow between the bot and the user. If the Dialogue Manager founds the query of the user is directly goes to the Natural Language Generation stage where user will be responded with a pre-defined answer. If the Dialogue Manager finds the query comes under the domain. The processed query along with the intents and entities will be used to search and fetch the relevant information from the Heriot Watt University website using the Search Engine. Then the results will be pre-processed after which we will be able to get data in a passages format and will be feed into the Question Answering System where our Vanila BM25 Algorithm will be able to extract the answer and source of the answer from the pre-processed data. Then the derived answers and the sources will be sent to the NLG and the output will be sent as a respond to the user's query.

The high-level overview of the chatbot workflow is shown in the figure 3.

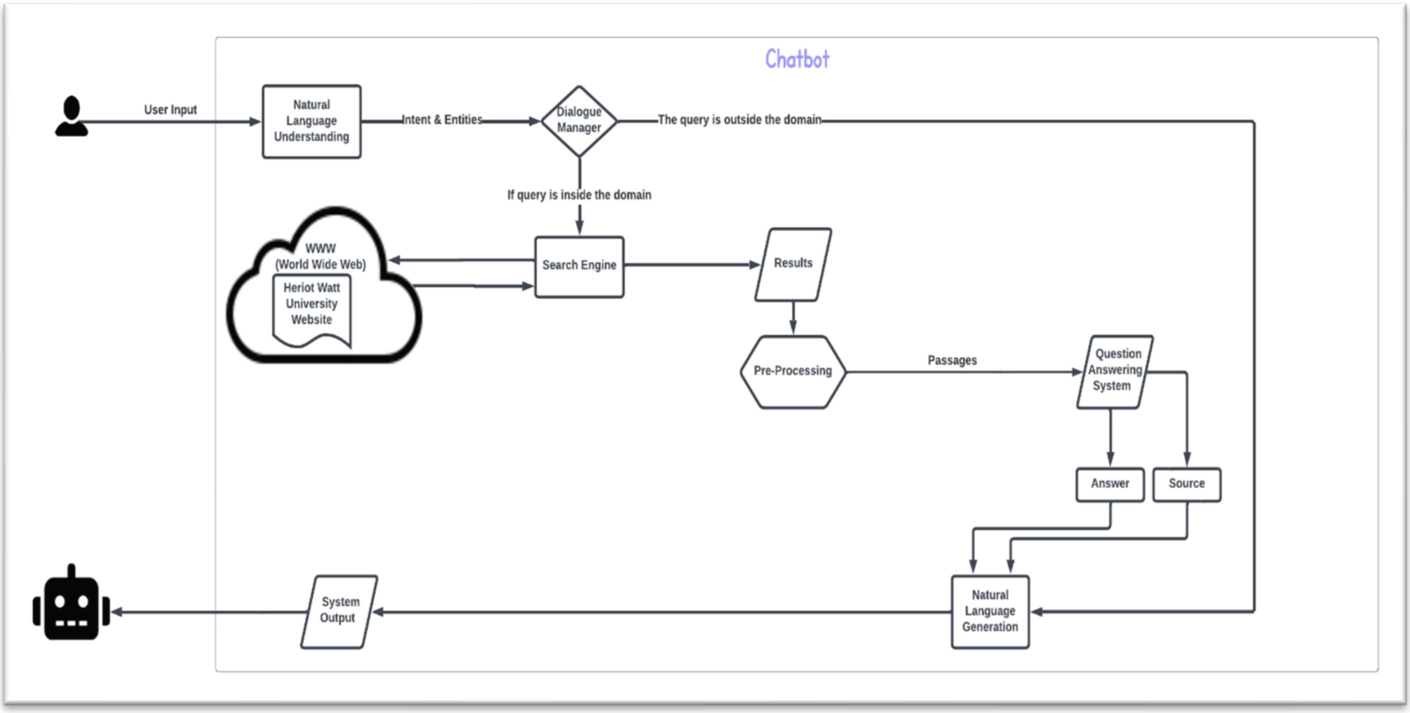


Figure 3: Workflow of the chatbot

2.6. Dialogue Systems:

A dialogue system, sometimes referred to as a conversational system or a chatbot, is an artificial intelligence (AI) system created to interact with humans in natural language. As for the Dialogue system, there will be a User Interface where the user will be able to send and receive text messages like the normal messaging software/Apps but instead of human sending a response, the model/software program will take care of the responding part by analysing the query and responding accordingly. Modern automated chatbots are often built using Machine Learning (ML) and Natural Language Processing (NLP) in the Artificial Intelligence (AI) field.

Here in our dialogue system, we will be using three different components namely Natural Language Understanding (NLU), Dialogue Management (DM) and Natural Language Generation (NLG). In the NLU part, the user's intent is categorised using a taxonomy that has already been defined, and from their utterances, the domain "entities," or the conditions necessary to fulfil that intent, are extracted from the user's input/query.

1. **Natural Language Understanding (NLU):** The "reading" phase is known as natural language understanding. NLU analyses text to ascertain the meaning and intent of what a client says by analysing sentence structure and intent. retrieving the domain "entities," or the specifications necessary to fulfil that intent, from the user's utterances. This is done by classifying the user's intent according to a previously established taxonomy.[12]

2. **Dialogue Management (DM):** The Dialogue Management mainly takes care of the dialogue flow between the user and the model/software. DM organises the flow of conversation to achieve the goal of the users' specified intent.[12] The dialogue management also has a specific module called Decision making module which contains a knowledge-Base (KB), from which the data which is required to fulfil the users' intent is retrieved and the data will be passed to the NLG. The dialogue management phase categorises the query type and establishes the appropriate category of responses the chatbot can use in response. Here in our entire NLP task, machine learning can be used for various purposes like, to find the intent of the query, to find whether the intent is inside the domain, etc.,
    - **Machine Learning (ML):** Natural Language Processing and ML have become crucial tools for handling free-form, unstructured text. Depending on the data source, both supervised and unsupervised techniques have a place. [10] To implement NLU and NLG, various machine learning methods were provided in several research articles. For Example, one of the popular methods for Q&A problems (like text classification, sentiment analysis) is Deep Neural Networks, division of Deep Learning (DL). [11] Deep Learning is the subset of ML, the use of Neural

Networks (NN) in conversational systems is a great breakthrough for creating powerful generative chatbots.

3. **Natural Language Generation (NLG):** The "writing" phase is considered as the Natural Language Generation. Basically, NLG catches the users' intent, and it creates the response for the intent. This includes grammatical structuring, content analysis, Language presentation etc., So far, Grammar rules or "canned text" have typically been used as the foundation for response generation. The "canned" approach is the standard for industrial conversational systems, even though recent generative models (Recurrent Neural Networks, commonly known as RNN) have greatly benefited natural language generation from a research perspective. This is because it is crucial for content experts to ensure that appropriately crafted responses are returned. [12] But in our case, we are using information retrieval method for a closed domain, As the dataset will be small size, I consider template-based answer delivery is the most convenient and efficient method for our chatbot.

2.7. Open Domain vs Closed Domain:

Open domain and closed domain are the two different methodologies in implementation of Conversational Understanding System (CUS). A Closed domain chatbot only has knowledge of one/few specific topics and only can generate a response from the knowledge/information that chatbot has. Whereas an open domain chatbot has knowledge on wide range (almost everything on internet) of domains/topics. Thus, an open domain chatbot will be able to answer wide range of answers. [13] Some of the examples for closed domain chatbots are:

- The domain of ELIZA's (the first chatbot) conversation was focused on the user's emotional state.
- Rose: Its domain is focused on the knowledge related to the products or services it is supporting.

- Woebot: Its domain is focused on the knowledge of human mental health like handling anxiety, depression, and stress.

Advantages of Open Domain:

- We have huge amount of data.
- Chatbot will be able to provide response for most of the questions/queries user asks.

Disadvantages of Open Domain:

- Uncertainty of handling questions, if the exact same words are used in different domains, chatbot may not provide correct answer.
- Need high computational power.
- Less accuracy.

As Closed domain chatbots only uses the knowledge on a selected domain to interact with the users, they are also known as domain specific chatbots.

Advantages of Closed Domain:

- Chatbot only needed to provide answers/responses about the selected domain.
- Less amount of data required.
- Require Less computational power.

Disadvantages of Closed Domain:

- Limited functionality.
- Lack of flexibility: unable to adapt to changes in user behaviour.
- Limited ability to understand context.

2.8. Task Oriented chatbot vs Social Chatbot:

Task oriented chatbots are the chatbots which are programmed to handle only one purpose. Task-oriented chatbots concentrate on doing specific tasks that are closely related to real-world activities, such as making hotel reservations, flight reservations, tracking orders and frequently asked questions on a specific domain area. [14] Task-oriented chatbots are often considered as useful resources for reducing wait times and improving productivity across a range of businesses. Task-oriented chatbots could also be integrated with backend systems, such as databases, API or web services, to get access to the data they require whenever they need it. Task oriented chatbots are also considered useful for letting customers execute tasks through a straightforward conversation interface rather than requiring them to visit and navigating through a complicated website or application in order to get the information, task-oriented chatbots can offer a more streamlined and practical user experience. They are able to run continuously and can lessen the requirement for human involvement.

Social chatbots are made to engage with people in a way that is more relaxed and conversational than other types of chatbots. Social chatbots are intended to engage in more open-ended interactions with users and offer them with information, help, or amusement, as opposed to task-oriented chatbots, which are focused on finishing certain tasks. Personal assistant chatbots like Siri and Alexa as well as chatbots used for customer service on social media sites like Facebook Messenger and Twitter are a few examples of social chatbots. Social chatbots are anticipated to display empathy, generate emotional responses, and strengthen relationships with users.[15]

Here as for our research purpose, we will be implementing a closed domain task oriented chatbot which will deliver answers for the users query i.e., Question Answering using the knowledge base.

2.9. Information Retrieval system:

Firstly, the user’s query will be processed, analysed and the requirement will be derived from the query, only then the information retrieval takes place with the help of the derived requirements.

2.9.1. Query processing:

Once the query was sent by the user to the chatbot, the pre-processing will be performed on the query i.e., stemming and Named Entity Recognition (NER)

- **Stemming:** The process of stemming involves transforming derivative words to their word stem, base, or root form. Often, this takes the form of a written word, such as "ing", "ly", "es", or "s".[16]
- **NER:** The named entity recognition method is used to recognise and extract from the text entities like individuals, groups, places, dates, and other significant information. Here in our case, we will be able to extract course name, course code, etc., from the query.

2.9.2. Information Retrieval techniques:

Some of the advanced models/algorithms are, BERT[17] (Bidirectional Encoder Representation Transformer) model, GPT-3[18] (Generative Pre-trained Transformer 3) model, RoBERTa[19] (Robustly Optimised BERT Approach), GloVe[20] (Global Vectors for Word Representation), Vanilla BM25[21].

Among these, I consider three well-suitable techniques in the area of information retrieval and representation. They are Vanilla BM25, BERT, and GloVe. Each of these approaches has advantages and disadvantages, and the best approach to adopt will rely on a number of factors, including the project's unique needs, the resources at hand, and the qualities of the data being utilised.

A well-known and long-standing probabilistic information retrieval model is the Vanilla BM25. It is a straightforward yet efficient technique for sorting documents according to the frequency of terms and inverse document frequency, and it is well-known for its capability to handle enormous document collections. Since Vanilla BM25 does not need pre-training or fine-tuning, implementing it in a project is not too difficult.

Modern NLP model BERT (Bidirectional Encoder Representations from Transformers) has attracted a lot of interest lately. It is a pre-trained language model that collects contextualised word embeddings using a transformer-based architecture. BERT has demonstrated outstanding performance on a variety of NLP tasks, including text categorization, named entity identification, and question answering. It is notable for its capacity to capture complex language representations, including semantic meaning and syntactic structure.[22]

Another popular technique for word embedding is GloVe (Global Vectors for Word Representation). The approach learns word vectors from co-occurrence statistics in a large text corpus in a static, count-based manner. Although BERT embeddings are contextualized, GloVe embeddings are not recognised for their capacity to capture word meaning.[20]

BERT and GloVe are known to perform better than Vanilla BM25 in many NLP tasks, particularly those that rely on understanding of the meaning and context of words. In particular, BERT has shown progressive outcomes in several standard datasets, and it has also grown to be a preferred model for many NLP applications.

However, despite their outstanding effectiveness, BERT and GloVe have several drawbacks. Their computational complexity and resource requirements are a significant constraint. Due to its size and complexity, BERT uses a lot of memory and processing resources during both the pre-training and fine-tuning steps. GloVe, on the other hand, is based on static embeddings learnt from co-occurrence data and might not be as successful at capturing nuanced contextual information as BERT.[22] [20]

In comparison, Vanilla BM25 is a far less resource-intensive approach to execute. It does not require substantial pre-training or fine-tuning and is efficient and effective in processing massive document collections. Vanilla BM25 is a viable option for applications with minimal resources or time limitations since it is simple to build in Python using libraries. [21]

It has been determined that Vanilla BM25 is the best option for the information retrieval task based on the unique needs of our research project, the resources available, and the properties of the data being employed. One of the main reasons for making this choice was the ease of deployment, effectiveness in managing big document collections, and fit for projects with limited resources or time restrictions.

In conclusion, Vanilla BM25 is a practical and efficient option for information retrieval, taking into account criteria such as simplicity, efficiency, and resource needs, even if BERT and GloVe are sophisticated NLP approaches that give greater performance in many situations.

2.10. Question Answering

Question Answering is the ability of a chatbot/system to handle a human user asking certain questions and the ability of the chatbot/system to provide a human understandable answer which is retrieved/generated using some kind of data source/database. There are many varieties of Question Answering, but extractive QA is considerably accurate compared to generative QA, because it includes asking questions whose answers can be determined by a section of text in a document, which could be a web page, a formal agreement, or a news story.

Many contemporary QA systems, such as semantic search engines, intelligent assistants, and automatic information extractors, are built on the two-stage method of first finding the most relevant documents from dataset and then extracting responses from them.[22] Question Answering models are considered really useful that they allow chatbot to analyse the intent of the users query, then retrieve and deliver the answers that will satisfy the questions/query that they have received from the user. The quality of Question Answering Systems is completely based on the dataset they use. In our case we will be using the Heriot Watt university's course details website as our dataset.

2.11. Types of chatbot responds:

2.11.1. Rule-Based responses:

Chatbots that operates corresponding to a set of rules and pre-specified responds are commonly known as rule-based chatbots. When chatbot faces any inquiries that are not specifically related to the domain for which they were developed may not be responded.[23]

- **Advantages:**
  1. Since they depend on predetermined norms and conditions, they are simple to build and manage.
  2. Effective for uncomplicated, easy activities like addressing FAQs.
- **Disadvantages:**
  1. They have limited functionality and are rigid because they can only reply to queries or inputs that they have been configured to handle.
  2. Unable to handle difficult queries or input in natural language.

2.11.2. Retrieval based responses:

These Chatbots employ machine learning techniques to find the best possible response for a given input after being trained on massive volumes of data. Based on prepared replies and keyword recognition methods, they operate.[24]

- **Advantages:**
  1. Has the ability to respond effectively to a variety of questions by finding the most pertinent information in a predefined database.
  2. They can be improved by training on a big dataset.
- **Disadvantages:**
  1. They are constrained by the quality and size of the training database.
  2. Might have trouble answering more challenging questions that call for in-depth contextual knowledge.

2.11.3. Generative responses:

These Chatbots do so using deep learning methods like neural networks. They are able to provide replies that are human-like and comprehend the context of the dialogue. They are not constrained to predetermined replies and have the ability to independently produce new responses. [25]

- **Advantages:**
    1. Has the ability to process difficult questions and produce customized replies using machine learning algorithms.
    2. Can get better over time with more data and feedback.
- **Disadvantages:**
    1. Need a lot of data and processing power to continue functioning and train.
    2. Without sufficient supervision, responses could be inappropriate or flawed.

It is clear from the list of benefits and drawbacks that every type of responses has certain advantages and drawbacks. For straightforward activities like FAQs, rule-based replies are straightforward and efficient, but retrieval-based solutions are appropriate in situations when a pre-defined database of information is provided. Although more flexible and capable of delivering answers for complicated queries, generative replies need more training data and processing resources. Based on its benefits and drawbacks for the intended application, the implementation of Vanilla BM25, a retrieval-based approach, may be a good option when taking into account the unique requirements and restrictions of our chatbot.

2.12. Challenges in Chatbots:

i. **Google API integration**: Understanding and putting into practise the API documentation, taking care of authentication and authorisation, and controlling API rate restrictions may be necessary when integrating with the Google API for search features.

ii. **Web scraping**: Web scraping may entail dealing with dynamic web sites, parsing HTML, collecting pertinent data, dealing with CAPTCHAs, and making sure that website terms of service are followed.

iii. **Data quality and reliability**: Finding appropriate data for our chatbot can be challenging when dealing with noisy or incomplete data, which can affect the reliability and quality of the data gathered via web scraping.

iv. **Natural language understanding**: Due to variations in expressing same sentences in various manner, Grammar and user intents, accurately comprehending and interpreting users' natural language queries can be difficult and requires techniques like intent detection, entity recognition, and context awareness.

v. **Information retrieval**: To achieve precise and meaningful results, retrieving pertinent information from the web page retrieved by web scraping may need techniques like keyword extraction, document rating, and relevance score.

vi. **Natural language generation**: It may be necessary to use strategies like template-based replies, response variety, and language generation models in order to produce responses that are grammatically correct, contextually relevant, and engaging.

vii. **Dialogue Management**: To guarantee fluid and understandable user interactions, it may be difficult to handle user interactions, manage conversation flow, handle errors, and provide appropriate error messages or prompts for user clarification.

viii. **Maintenance and updates**: The long-term maintenance of the chatbot could cause difficulties, including handling changes to the Google API or website structure/design as well as routinely updating and maintaining the chatbot to ensure the relevance, accuracy, and trustworthiness of responds.

ix. **Data privacy and security**: Implementing appropriate security measures, processing user data securely, and maintaining compliance with important privacy laws may be necessary to guarantee the privacy and security of user data and to follow to data protection rules.

x. **Performance optimization**: It may be necessary to carefully evaluate the system's architecture, algorithm optimizations, and resource management in order to optimize the chatbot's performance in terms of response time, scalability, and efficiency.

These are some of the challenges we may encounter while implementing our closed domain chatbot.

3. Professional, Legal, Ethical, and Social Issues

3.1. Professional Issues:

Python will be used as the primary programming language and Anaconda IDE as the project's platform throughout.

The written code will be tested, written to a high standard, well commented throughout to provide more clarity on what it is. And the implementation of the code will follow British Computing Society's code of conduct. The dataset will be used under Data Protection Law Policy. There will be sufficient documentation provided for the work. Any third-party software, libraries, or additional products will only be used if permissible by their licensing. Any citations or external data will be properly cited.

3.2. Legal Issues:

This research project strictly complies with the data security policies of the local government (United Kingdom). As mentioned earlier, all third-party libraries licences will be respected, and included into project files.

3.3. Ethical Issues:

This research project requires human evaluation experiment, any conversation between the user and the chatbot shouldn't have any kind of personal information which can lead to the participant, ensuring no privacy breach. All dataset i.e., the websites we will be using are publicly available on [Heriot Watt University website](). Thus, there is no risk of violating any ethics code.

3.4. Social Issues:

The expansion of employing the project's models in a real spoken dialogue system for conversational agents in the future could also pose few problems. Some of them are:

1. The chatbot could deliver wrong information.
2. It might respond to abusive language.

This will result in a terrible user experience. If any of these problems are found, the system will be modified accordingly.

4. Project Plan

4.1. Gantt Chart

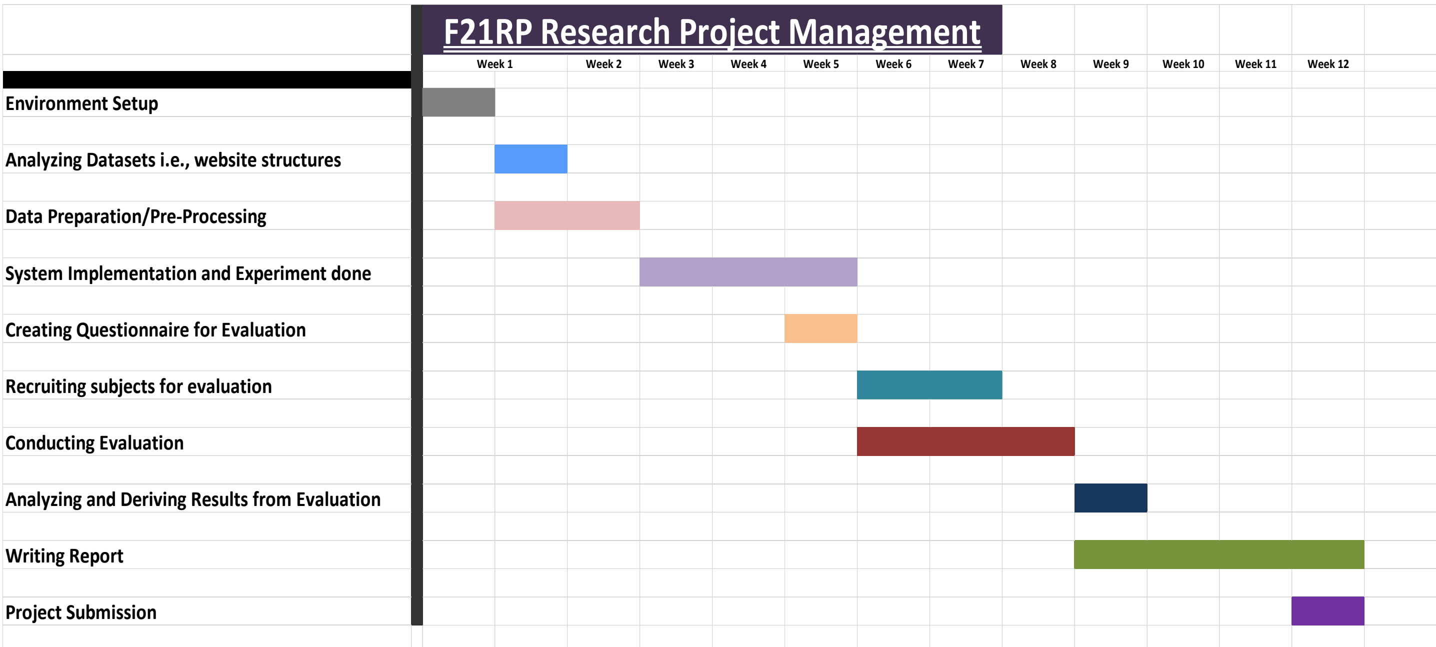


Figure 4: Gantt Chart

The figure 7 shows the plan of this project. This research project is planned to be done in 12 weeks period. As we can see in the figure 7, we will be recruiting people for evaluation but also, we will be conducting evaluations at the same time [week 6 & 7]. The implementation of this project includes Analysing datasets, Pre-processing data, System Implementation & testing, Creating Questionnaires, recruiting people and conducting evaluation, Analysing the evaluation results, writing report.

5. Requirement Analysis

The outcome of this research project will provide us with clarification of what kind of information delivery of chatbots leads to greater user satisfaction and improves reliability on chatbots. For evaluation of this research, A prototype chatbot will be implemented which delivers university's course details and below are some fundamental specifications, broken down into functional and non-functional requirements.

5.1. Functional Requirements:

ID	Details	Type	Priority
R1	The Chatbot must be able to provide information from the university's course website	• Information Retrieval	MustHave
R2	**Chatbot Responses**		
R2-A	The Chatbot must be able to deliver answer for the query	• Information Delivery	MustHave
R2-B	The Chatbot must be able to deliver sources containing the answer	• Information Delivery	MustHave
R2-C	The Chatbot must be able to deliver answer along with sources	• Information Delivery	MustHave
R3	The Chatbot must be able to understand users' natural language input	• Natural Language Understanding	MustHave
R4	The Chatbot should be able to handle users query when it gets out of the domain	• Dialogue Management	CouldHave
R5	The Chatbot should be able to deliver the retrieved information in a proper user understandable manner	• Natural Language Generation	ShouldHave
R6	The Chatbot should be able to keep log of each and every user's query and its responses	• Logging	ShouldHave

5.2. Non-Functional Requirements:

ID	Details	Type	Priority
R1	The chatbot should be accessible on different devices like Computers and Mobile phones	• Compatibility	CouldHave
R2	The Chatbot should provide accurate and reliable information to the user	• Reliability	ShouldHave
R3	The Chatbot should provide response quickly	• Performance	CouldHave
R4	The Chatbot must not collect any of the user's personal data	• Security	MustHave

6. Implementation

6.1. Development environment setup

To guarantee the smooth integration of several technologies and tools, the development environment for this chatbot project was carefully configured. I picked Python 3.9.0 as our primary programming language because of its extensive library and active community, particularly in the fields of web scraping and machine learning (ML). We were able to construct separate environments since the whole development process was managed within Conda, a potent package and environment management system. The avoidance of dependency conflicts and the maintenance of consistent behavior across various development configurations were made possible by this isolation. Because of its sophisticated code editor, debugger, and broad support for Python's web frameworks and libraries, PyCharm was chosen as the main Integrated Development Environment (IDE).

The chatbot's user interface, which offers an engaging and user-friendly experience, was created with the help of streamlit. Streamlit was a perfect fit as it was compatible with Python and made transforming data scripts into shareable web apps simple. Additional libraries that were added to the environment included BeautifulSoup for effective web scraping, Spacy for more complex NLP tasks, and Pandas for data manipulation. These libraries made sure that every facet of chatbot functionality—from data processing to user interaction—was effectively handled. This well-built development environment provided a strong basis on which the chatbot could be created quickly and effectively.

6.2. NLU Component

6.2.1. Dataset creation and Preprocessing:

A crucial stage in building the chatbot's natural language understanding (NLU) component was generating and preparing a dataset that met the project's unique requirements. The `data.py` file contains the carefully constructed dataset by the author [i.e., Jose Berlin]. As you can see in the figure 8, the data was first organized using a dictionary format (`dict()}), and it consists of a set of sentences that have been mapped to their respective intents. This represents the range of user questions that the chatbot is anticipated to be able to answer to. In order for the chatbot to correctly comprehend and classify user query, intents are essential.

```
# This Dataset has been completely created by Jose Berlin (i.e., the author) of this program
def declare_data(self):
    self.raw_data = {'I would like to talk about Msc AI': 'select course',
                     'I want to talk about Bsc computer science': 'select course',
                     'I want to talk about Bsc psychology': 'select course',
                     'I would like to talk about MPhys Physics': 'select course',
                     'Tell me about Msc Actuarial Management with Data Science': 'select course',
                     'Tell me about Msc Applied Petroleum Geoscience': 'select course',
                     'I want to talk about MPhys Physics': 'select course',
                     'How about Msc': 'select course',
                     'How about Bsc': 'select course',
                     'How about MEng': 'select course',
                     'How about Beng': 'select course',
                     'Bachelors': 'select course',
                     'Masters': 'select course',
                     'post graduate': 'select course',
                     'postgraduate': 'select course',

                     'How are the courses delivered?': 'delivery',
                     'Is this a hybrid course?': 'delivery',
                     'what is the course delivery for this course?': 'delivery',
                     'Can I take the courses remotely?': 'delivery',
                     'What is the mode of course delivery for international students?': 'delivery',
                     'Do you offer part-time in this courses?': 'delivery',
```

Figure 5 Sample of the query-intent dataset created by the author

This dictionary was then converted into a Pandas DataFrame for quick processing and user-friendliness, which was inspired by the DataFrame's capacity to manage massive datasets in a structured manner. 'Sentence' and 'Intent' are the two main columns of the DataFrame we created. The user statements or queries are stored in the 'sentence' column, and the corresponding intent labels are stored in the 'intent' column. In addition to streamlining the first part of the machine learning pipeline, this organized approach to dataset construction and preprocessing also made it easier for later phases, such model training and assessment, and guaranteed that the NLU component operates with high accuracy and efficiency.

6.2.2. Intent Classification:

6.2.2.1. Model Selection and training:

A key decision in the project was which model to use for the chatbot's intent recognition. The Neural Network, Support Vector Machine (SVM), and Logistic Regression models were the three that were assessed. An 80/20 split was used to separate the dataset into training and testing sets. The Neural Network performed poorly for this application after it was trained for 100 epochs, with a test accuracy of about 50%. After testing, the SVM model produced results with a higher accuracy of around 71% and a roughly 76% 10-fold cross-validation accuracy. Nonetheless, the Logistic Regression model produced the most encouraging outcomes. With a 10-fold cross-validation, the accuracy of this model increased to around 78%, from about 74%.

Utilizing the spacy en_core_web_sm model, stemming was added for even more improvements. With this addition, the accuracy of the Logistic Regression model increased to 86.21% from 78% without stemming. It is noteworthy that the stemming procedure resulted in a significant increase in processing time, so posing a trade-off between efficiency and accuracy. The table below shows the accuracy achieved during the testing period of different Models.

Model	Test Accuracy	10-fold CV Accuracy	With Stemming Accuracy
Neural Network (NN)	~50%	-	~`52.86%
Support Vector Machine (SVM)	~71%	~76%	~80.12%
Logistic Regression	~74%	~78%	~86.21%

Figure 6 Different Models used for Intent recognition and their accuracy with 10-crossfold and Stemming

6.2.2.2. Performance Evaluation:

To determine which model was best for intent identification, the models' performances were thoroughly assessed. The accuracy of the Neural Network trailed behind, however SVM and Logistic Regression both produced encouraging results. But the higher performance of Logistic Regression—especially after stemming—made it stand out. Despite requiring more processing time, stemming improved accuracy from 78% to 86.21%, demonstrating how well this method captures the subtleties of spoken language. This is a critical development for a chatbot, as it is essential to correctly determine the purpose of the user in order to come up with responses. However, stemming adds a substantial processing time, which emphasizes the necessity for a balanced approach between accuracy and reaction time in real-world chatbot applications.

6.3. Chatbot Response Strategies

6.3.1. Types of response: Implementation

An essential component for the accomplishment of this research is the chatbot's implementation of the many answer kinds. 'Information with sources' is how the chatbot responds by default. This default configuration was selected to deliver a thorough answer style that includes the sources for additional reference and trust-building along with to the required details.

As we can see in figure 10, by utilizing specific keywords, we may change the response that users get. For example, "/sys1" instructs the chatbot to offer information solely, "/sys2" directs the bot to deliver information combined with sources (which is the default mode), and "/sys3" causes the chatbot to present simply the sources. This part of the chatbot's functioning depends substantially on the Dialogue Manager (DM). When the DM detects specific keywords, it modifies the response mechanism appropriately and tells the NLG, how the response should be constructed.

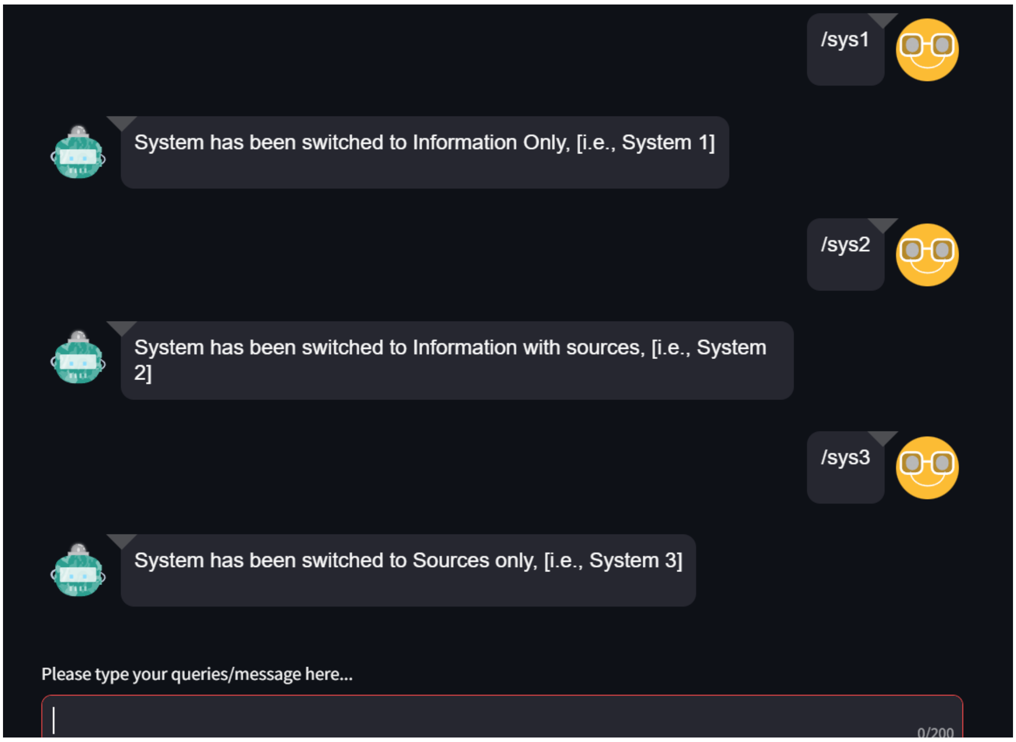


Figure 7 Switching between different responses using keywords.

6.4. Web scraping component

6.4.1. Implementation of web scraping:

A major component in the chatbot's ability to retrieve course-specific data from the Heriot-Watt University website is its application of web scraping. The NLU recognizes "select course" as the user's query intent, at which point this procedure begins. The DM analyzes results from the official Heriot-Watt University website by doing a focused search inside the [Heriot-Watt University website]() using the Google search. This focused strategy guarantees that the search results are limited to the official university website and pertinent. The DM gives control to the web_scraping class along with the obtained site URL after determining the correct website address.

The fundamental part of the information extraction procedure is the web_scraping class. It starts by utilizing the requests library to retrieve the raw HTML content after being initialized with the precise website URL. The class circumvents any access limits or limitations that websites could have against bots by sending an HTTP GET request with a user-agent header to simulate a browser request and guarantee successful page retrieval.

Get_abs_url, a crucial method in the class, is responsible for translating relative URLs into absolute ones. The urljoin method from the urllib.parse module is used to accomplish this, ensuring that all links taken from the webpage are whole and operational. The get_passages method encapsulates the main function of the class. Using BeautifulSoup, this technique parses the HTML content and methodically finds and extracts text and material under different HTML headings, such h1, h2, and so on. Paragraphs, lists, tables, and description lists receive special attention from it, which guarantees a thorough extraction of pertinent data.

Furthermore, the technique is designed to locate and retrieve links, sometimes nested inside anchor tags (<a>), that pertain to course applications or contact details. When these connections are found, they're added to the list of texts along with helpful titles like "Apply" or "Contact," which makes the content more beneficial. To maintain the integrity of the information structure, a default message is inserted in situations when particular connections cannot be located.

An ordered collection of information combined into a structured list of dictionaries is the result of this approach. Because each dictionary has a heading and its accompanying content, the scraped data is simply available for subsequent processing and is also structured. The web scraping has been carefully developed to tackle typical issues like managing relative URLs and navigating various HTML structures, guaranteeing the extraction of valuable and organized material from the Heriot-Watt university website.

6.4.2. Data storage and Management:

Efficient data management and storage are critical to the chatbot's implementation, particularly once data is retrieved via web scraping. The last critical step is safely saving this data to provide smooth access and interaction inside the chatbot interface after the web_scraping class has extracted the pertinent parts from the Heriot-Watt University website. under the Streamlit framework, the scraped data is kept for this reason under st.session_state.data.

It was a purposeful decision to save the data in st.session_state.data. The session state feature of Streamlit offers an easy and effective approach to save data between program repetitions. This method works especially well in web-based chatbot environments where data durability is essential and the user interface is dynamically changed. The chatbot makes sure that once the data is received, it stays available for the duration of the user's session by keeping the scraped data in the session state. This eliminates the need for repetitive scraping. By cutting down on pointless server queries, this technique not only improves the chatbot's performance but also offers a more seamless and responsive user experience. It successfully tackles the problem of data persistence in a setting where user activities may cause the program to be launched more than once. This kind of data management demonstrates how well web scraping capabilities are integrated with Streamlit's powerful data handling features, guaranteeing the effectiveness as well as dependability of the chatbot in providing precise and current information.

```python
import streamlit as st
from streamlit_chat import message as st_msg
# from streamlit_webrtc import webrtc_streamer
import random
import requests

from DM import DM

class main:
    def __init__(self):
        self.dm = DM()
        self.data = []
        self.UI()

        if "data" not in st.session_state:
            st.session_state.data = dict()
            st.session_state.site = ""
            st.session_state.sys_no = 2
```

Figure 8 Initializing the session to handle the volatility of data

6.5. NLG Component

6.5.1. Passage Retrieval and Ranking:

The chatbot's effective and vital passage retrieval and rating system makes sure that all responses it provides are correct and pertinent to the user's aim. Tokenization, the act of dividing a user query into discrete tokens, is the initial step that takes place when the query is received. The saved passages are then retrieved from st.session_state.data, which contains the pre-processed web-scraped data, using these query tokens as the foundation.

The BM25 model is used to determine how relevant each passage is to the given query. This model is widely used in the field of information retrieval and is well-known for its ability to rank documents based on their relevance. It will determine scores for each passage in relation to the query tokens by using the terms of word frequency and inverse document frequency.

The BM25 scores are turned into a NumPy array after they have been calculated. By utilizing NumPy's powerful computational capabilities, this translation makes handling and processing of the scores easier. The next stage involves determining which passage is the most pertinent, which is accomplished by using the np.argmax function to get the index of the paragraph with the greatest BM25 score.

Next, the passage from the data attribute that corresponds to this index is taken out. This paragraph is provided as the answer because it is the one that is thought to be most relevant to the user's query. The chatbot makes sure that the information it offers is not only contextually appropriate but also specifically customized to the user's question by using the BM25 algorithm for retrieval and ranking. This method highlights the chatbot's capacity to sort through a multitude of data and identify the most appropriate response, demonstrating an efficient use of sophisticated retrieval techniques in a useful, user-focused environment.

6.6. Streamlit UI integration

6.6.1. UI Design and Layout:

To provide a pleasant and user-friendly experience, the chatbot project's User Interface (UI) design and layout are essential. The application's main class is in charge of configuring the user interface (UI) using a specific UI method and starting the dialogue management system (Dialogue Manager, DM). The main feature of the interface is a chat window, which gives users a comfortable and engaging way to interact with the chatbot.

The UI welcomes users with a title and introductory text at the beginning, which are created using Streamlit's st.title and st.write functions. This introduction, which provides crucial details about the capabilities and goal of the chatbot, establishes the tone for the user experience. The chat history container, which is made with st.container, is a crucial component of the user interface. This feature helps users follow the development of the engagement by storing and showing the history of the ongoing discussion.

Messages within the chat are displayed using st_msg from streamlit_chat, an effective tool for creating chat-like interfaces within Streamlit applications.

This approach provides a realistic chat environment, contributing to a more natural and engaging user interaction.

User inputs are captured through a text input field, implemented with st.text_input. This field serves as the primary means for users to enter their queries or messages. When a user submits input, it activates the got_msg method, which is responsible for processing the user input and generating the chatbot's response. Importantly, the got_msg method also updates the chat history in the Streamlit session state, thereby maintaining the flow and continuity of the conversation.

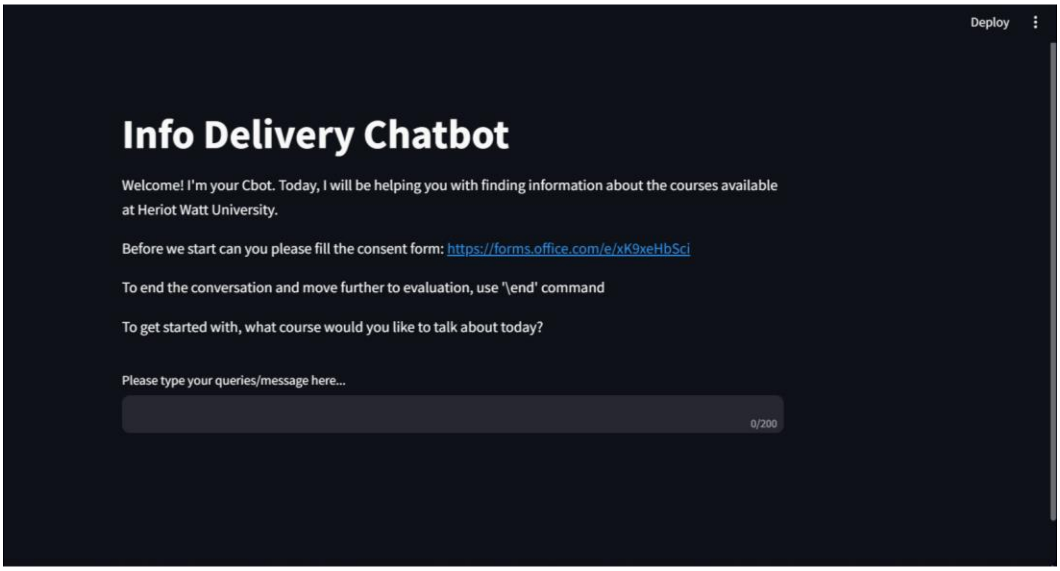


Figure 9 Chatbot web UI

Furthermore, the UI includes a consent form link and clear instructions for terminating the conversation. This aspect underscores the importance of user consent and provides users with straightforward guidance on how to navigate and conclude their interaction with the system. Additionally, to enhance the user experience, the application employs JavaScript to ensure that the chat window automatically scrolls to display the latest message. This feature keeps the most recent part of the conversation in view, facilitating a smooth and uninterrupted chat experience. Overall, the UI design and layout are carefully crafted to be intuitive, user-centric, and conducive to an efficient and pleasant chatbot interaction.

6.7. System testing and evaluation

6.7.1. System Methodologies:

The development of the chatbot system was spearheaded solely, employing the Agile software development methodology. Agile, known for its flexibility and iterative approach, was particularly suitable for this project due to its adaptive nature and emphasis on incremental progress. This methodology facilitated frequent assessment and adaptation, allowing for continuous refinement of the system throughout the development process. The implementation was structured in sprints, each focusing on specific features or improvements, thereby ensuring that the project progressed in a manageable and structured manner. This approach not only allowed for consistent progress checks and immediate rectifications but also ensured that the chatbot was developed in a way that could readily adapt to emerging requirements. The use of Agile, with its user-centric focus and iterative development cycles, significantly contributed to the creation of a chatbot system that was both responsive to user needs and adaptable to the evolving scope of the project.

6.8. System Deployment

6.8.1. Local Deployment:

Local deployment served as the initial phase, where the chatbot was deployed on a local server to undergo rigorous testing and evaluation. This stage was essential for identifying any bugs, performance issues, or areas needing improvement in a controlled environment. The deployment process involved setting up a local server environment, configuring the necessary dependencies, and launching the Streamlit application. The chatbot was then interactively tested to ensure all components, including the NLU, DM, web scraping, and response generation systems, worked harmoniously and efficiently. This phase also provided an opportunity to fine-tune the chatbot's performance, ensuring optimal response times and accurate information retrieval.

6.8.2. Challenges and Solutions:

One of the significant challenges encountered in the project was maintaining the accuracy and relevance of the web-scraped data. This issue was particularly challenging due to the dynamic nature of web content, which can frequently change. The solution was meticulous preprocessing of the scraped data to ensure its relevance and accuracy. Careful preprocessing involved validating and cleaning the data, ensuring it was up-to-date and relevant to the user's queries.

A major development in the project was the experimentation with the latest Large Language Models (LLMs) to enhance the NLU and NLG capabilities of the chatbot. This development was conducted in a separate GitHub repository, available at HW-Cbot. The repository outlines an advanced approach where the chatbot, referred to as 'Cbot', operates as an assistant delivering information about Heriot-Watt University courses. In this setup, the chatbot is designed to operate on a Linux system with Ollama installed, using a specific modelfile configuration for processing queries.

To streamline the information retrieval process, the "Download courses.py" script was implemented. This script automates the downloading of all available courses from the Heriot-Watt University website, storing them in the "HW courses" directory in a "coursename.txt" file format. Once the courses are downloaded, the "QA.py" script takes over. This script first identifies the relevant course based on user input, utilizing the "sentence-transformers/all-MiniLM-L6-V2" model for finding the most pertinent course file. Then, when users pose questions about the selected course, the relevant text file is broken down into chunks. The TfidfVectorizer() identifies the most relevant chunk for the user's query, and the Ollama model is then employed to generate an accurate and contextually relevant answer.

Despite these advancements, a notable setback was encountered with the intended deployment on the Heriot-Watt University Linux server. Due to difficulties in installing essential dependencies such as transformers and

Streamlit, and the inability to resolve these issues in a timely manner, the project had to revert to the base model that primarily utilized the BM25 algorithm. This fallback, while not ideal, ensured that the project remained operational and continued to serve its primary function effectively. The experience highlights the importance of adaptability and the ability to pivot in response to technical challenges in software development projects. The advanced version of the chatbot, along with its development journey, can be explored further at the provided GitHub repository link [HW-Cbot].

7. Evaluation and Results:

7.1. Experiment Design:

The experiment was carefully designed with a view to assessing the effectiveness and user experience of chatbot systems which were created in order to provide university course information. This study involved 14 students from Heriot-Watt University, who were all be asked to use one of the three different chatbot systems. To assess the chatbot's efficiency, user satisfaction and overall experience, an experimental design has been established that aims to collect qualitative as well as quantitative data.

7.1.1. Participant Selection and Preparation:

The participants were selected from Heriot-Watt University's student body to guarantee that they reflected the targeted user base for the chatbot. Before engaging with the chatbot, every participant had to fill out a consent form. This form addresses any questions they may have had about privacy and data usage, as well as the nature of their involvement in the study. Ensuring ethical compliance and instructing participants about their rights and the study's scope required undertaking this crucial step.

7.1.2. Instruction and Interaction:

After completing the consent form, participants were given an instruction card. This card contained specific queries and tasks that they were to perform using the chatbot. These tasks were designed to cover a range of functionalities and response types of the chatbot, allowing a thorough evaluation of its capabilities. The instructions were crafted to guide the participants through a series of interactions that would elicit a comprehensive range of responses from the chatbot, ensuring a thorough assessment of each system.

7.1.3. Chatbot Systems:

As I’ve mentioned earlier, the chatbot systems were designed to deliver information in three formats:

- System 1 [Answer Only]
- System 2 [Answer with Sources]
- System 3 [Sources Only]

Each participant interacted with one of these systems, randomly assigned, to ensure an unbiased evaluation of each system's effectiveness and user experience.

7.1.4. Evaluation and Feedback:

Upon completing the tasks outlined in the instruction card, participants were asked to fill out an evaluation form. This form was structured to capture their experience with the chatbot, focusing on aspects such as ease of navigation, accuracy of information, level of satisfaction, trust in the chatbot’s responses, and the likelihood of recommending the chatbot to others. The evaluation form consisted of a mix of Likert-scale questions and open-ended questions, allowing participants to provide both quantitative and qualitative feedback.

The design of this experiment was aimed at gathering detailed insights into each chatbot system’s performance from the perspective of the end-users. The data collected from this study will be instrumental in identifying strengths, weaknesses, and areas for future improvement in the chatbot systems.

7.2. Results:

The users interacted with the chatbot, indicating a preference for responses that included both answers and their sources. This choice suggests a user inclination towards verifying the accuracy of information independently, which is a significant aspect of building trust in automated systems like chatbots.

It's interesting to note that 11 out of 14 participants (or around 78.6%) answered positively when asked if they preferred chatbots over web page browsing. The relevance of preferring chatbots over webpages is further supported by this high favourable response rate.

In terms of the chatbot's accuracy in providing answers with sources, the ratings were predominantly high. This indicates that, 44% of the users found chatbot's responses to be quite accurate when the responses are delivered with sources. High accuracy is crucial for establishing the chatbot as a reliable source of information, especially in an academic context where correct information is essential.

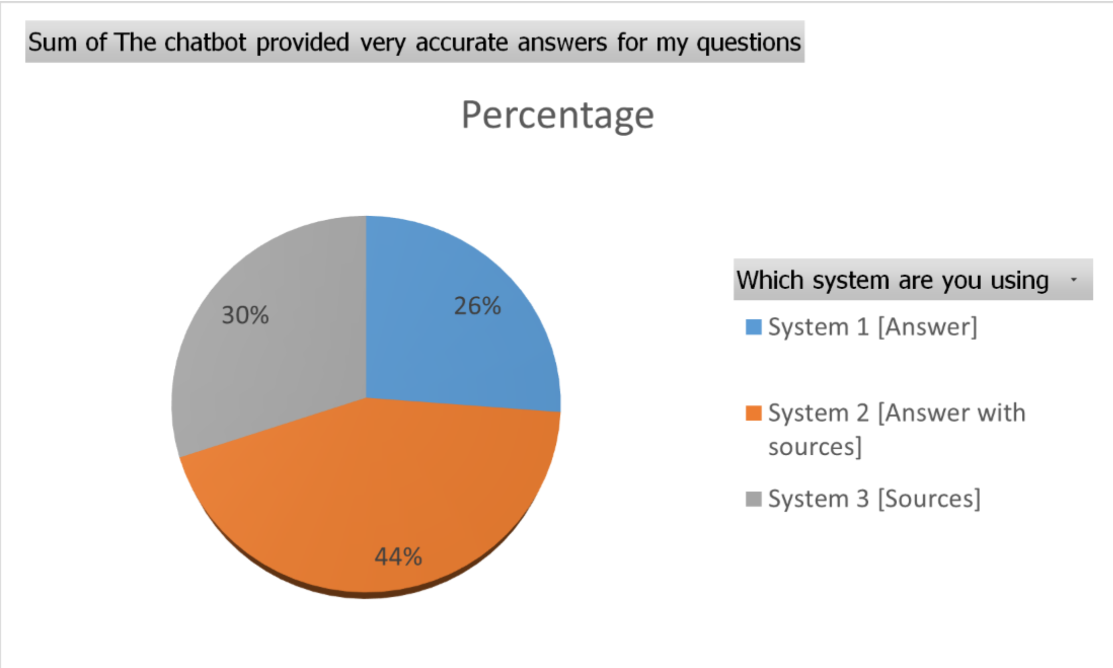


Figure 10 How much accuracy does the user's assume based on different system responses

User satisfaction also scored high when the information is delivered along with the sources. 49% of the users felt well satisfied when the sources were provided along with answers. This level of satisfaction is indicative of the chatbot effectively meeting the needs of the users. High satisfaction rates are likely to encourage repeated use of the chatbot for similar information retrieval needs in the future. But when the response only contains source, The user's gets the least satisfaction from using the chatbot.

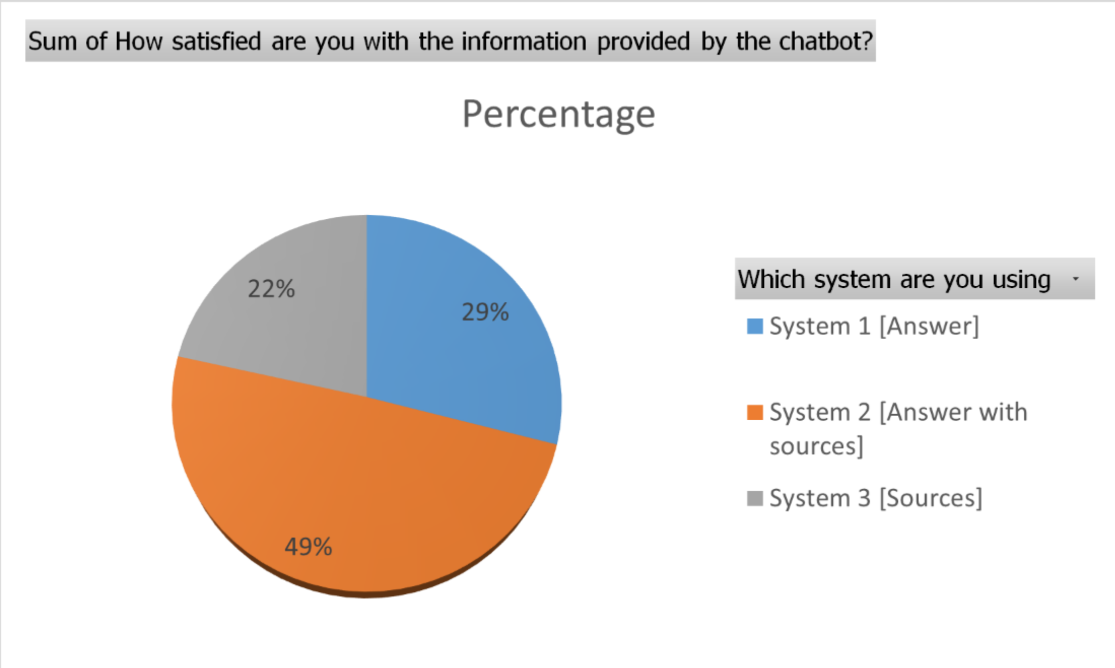


Figure 11 How satisfied the users are based on different responses by chatbot

When assessing the extent to which users trusted the answers provided by the chatbot, the responses were again favorably high, when chatbot delivers information with sources. This high level of trust is significant, as it reflects user confidence in the chatbot's capability to provide accurate and useful information.

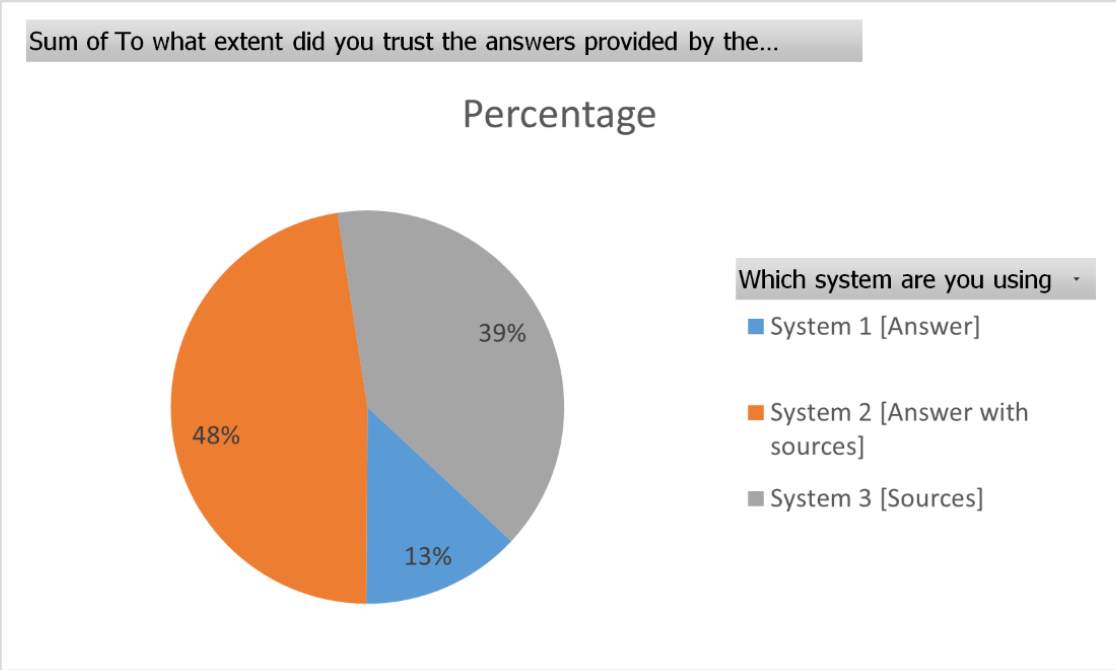


Figure 12 How much does the users trust the responses provided by chatbot.

Lastly, the likelihood of users recommending the chatbot to others for accessing university course information was evaluated. The scores were overwhelmingly positive, with most users willing to recommend the chatbot when the responses contain answer only and information with source. This willingness to recommend the chatbot to others not only highlights the system's utility but also its potential for wider adoption.

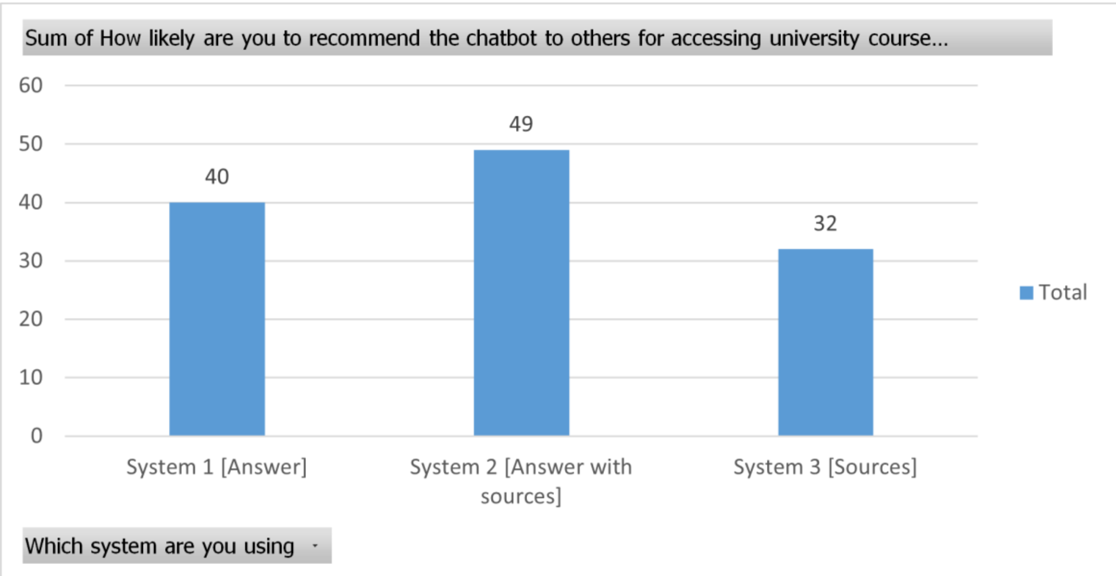
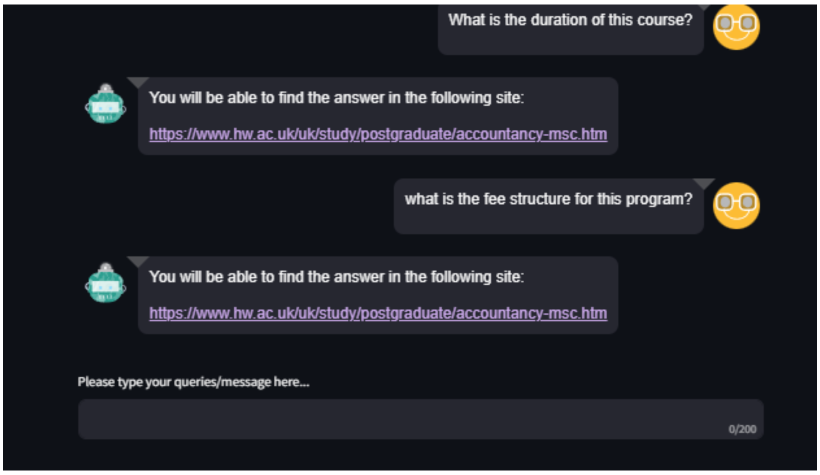


Figure 13 How likely are the users to recommend our chatbot on different responses provided.

Overall, the evaluation and results from the user feedback suggest that the chatbot system, particularly "System 2 [Answer with sources]", was successful in providing accurate, satisfying, and trustworthy information, leading to a high likelihood of recommendation by the users. These results are a strong indicator of the chatbot's effectiveness and potential as a reliable tool for accessing university course information.

(a)

(b)



(c)

Figure 14 Chatbot responses for different courses

8. Conclusion:

This research project aimed to assess the impact of different response formats (answer only, answer with sources, and sources only) on user trust, satisfaction, and recommendation likelihood in a chatbot system designed for accessing university course information. Drawing on feedback from 14 Heriot-Watt students, the project provided substantial insights into user preferences and the effectiveness of the chatbot in an academic setting. The findings indicate a pronounced preference among users for responses that include both answers and sources. This preference underscores the importance of not just providing accurate information but also supporting it with credible sources, enhancing the chatbot's reliability and trustworthiness.

In terms of accuracy, 44% of users found the chatbot's responses quite accurate when delivered with sources, emphasizing the crucial role of source inclusion in establishing the chatbot as a reliable information source. User satisfaction followed a similar trend, with 49% reporting high satisfaction when responses were paired with sources. This satisfaction level, however, dipped noticeably when the chatbot provided only sources, highlighting the need for a balanced approach in response delivery.

The trust placed in the chatbot's responses was also notably high when sources were included, reflecting a significant confidence in its capability to deliver not only accurate but also useful information. Furthermore, the likelihood of users recommending the chatbot was most positive when responses contained either answers only or answers with sources, suggesting that these formats were most conducive to user satisfaction and the chatbot's broader adoption.

The results of this evaluation, while indicating clear patterns, must be contextualized within the limitations of the study, particularly the relatively small number of test subjects. Despite this, the patterns observed lend strong support to the hypothesis that providing answers with sources leads to higher user satisfaction and trust. In conclusion, the project successfully demonstrated the potential of the chatbot, especially with the "Answer with sources" format, in providing an effective, trustworthy, and user-satisfactory experience in the domain of university course information retrieval.

9. Future Work:

The chatbot developed for this research project, while functional and effective within its scope, remains basic due to certain constraints. Time limitations played a significant role in defining the scope and complexity of the chatbot. Additionally, computing limitations of the personal computer used for development posed restrictions on the extent of features and capabilities that could be implemented. These factors collectively necessitated a more streamlined approach, focusing on core functionalities rather than extensive feature development. As a result, there exists ample room for enhancement and expansion in future iterations of the chatbot, allowing for more advanced features and capabilities to be explored and integrated.

- **Advanced Approaches for Natural Language Processing (NLP):** Using more complex natural language processing (NLP) techniques, such sentiment analysis and context-aware algorithms, might help the chatbot comprehend user inquiries better and provide more accurate responses.
- **Support in Multiple Languages:** As Heriot-Watt has various campuses in different countries, wider audience might use the chatbot if its multilingual support was expanded.
- **Scalability and Performance Optimisation:** Focusing on scalability and performance optimisation to guarantee that the chatbot can handle a greater number of concurrent user questions without sacrificing answer time or quality.
- **Custom Academic Information Delivery:** Developing the chatbot further, where a user's can login using their university email address first. Then, they would be able to get individualised academic information. With the use of this function, the chatbot would be able to provide precise information about a student's academic standing, including past performance, current grades, upcoming exam schedules, overall course progress etc.,

APPENDIX

Appendix A

Consent form

Consent form

* Required

User Evaluation Consent Form

Introduction: Thank you for participating in the evaluation of our chatbot as part of the research project "***User-Chatbot Interaction Analysis: A Comparative Study of Chatbot Experiences***". Before you begin, please carefully read the following information about the study and your participation. If you have any questions or concerns, feel free to ask.
Study Title: USER-SYSTEM INTERACTION ANALYSIS: A COMPARATIVE STUDY OF CHATBOT EXPERIENCES
Researcher(s): Jose Berlin Durai Yoseppu
Purpose of the Study: The purpose of this study is to evaluate user preferences and interactions with chatbot system designed to provide information from Heriot-Watt University's course website.
Study Procedures:

- You will be asked to interact with the chatbot under a scenario.
- The evaluation may involve completing questionnaires about your past experiences with chatbots and web surfing.
- You will also answer questions related to your experience with the chatbot and provide feedback.

Risks and Benefits: There are no anticipated risks associated with participating in this study. The benefits include contributing to the improvement of the chatbot's functionality and understanding user preferences.
Confidentiality: Your responses and personal information will be kept confidential. Your username or any other personal data won't be used for analysis, and only aggregated results will be analysed. Your username information will not be shared with third parties. This data will be stored only on a firewall-protected University server, or the University's secure encrypted cloud storage services (OneDrive). The data collected will be treated in accordance with your rights at all times and will be retained only for as long as is necessary for the purposes of evaluating the system and publishing the research. Further information, including details of who to contact if you have any concerns about how Heriot-Watt processes your data, is available here (https://www.hw.ac.uk/uk/services/docs/information-governance/PrivacyNoticeResearch□V4Finalversion.pdf)
Voluntary Participation: Participation in this study is entirely voluntary. You may choose not to participate or withdraw at any time without penalty.

1. HW Username [Eg: aa2023] *

Enter your answer

2. By clicking "I Agree" below, you acknowledge that you have read and understood the information provided in this consent form. You voluntarily agree to participate in this study. *

- I Agree
- I disagree

Appendix B

System evaluation forms

Note: One of these scenarios will be provided to a user to interact with chatbot

1. **Ask to talk about Artificial Intelligence MSc:**

   Ask the following questions about the course:

   - Overview
   - Type of course delivery
   - Entry requirements
   - Duration
   - How to apply

2. **Ask to talk about Business Analytics MSc:**

   Ask the following questions about the course:

   - Overview
   - Course contents
   - Program fees
   - Contacting faculty
   - Type of course delivery

3. **Ask to talk about Computer Science BSc (Hons):**

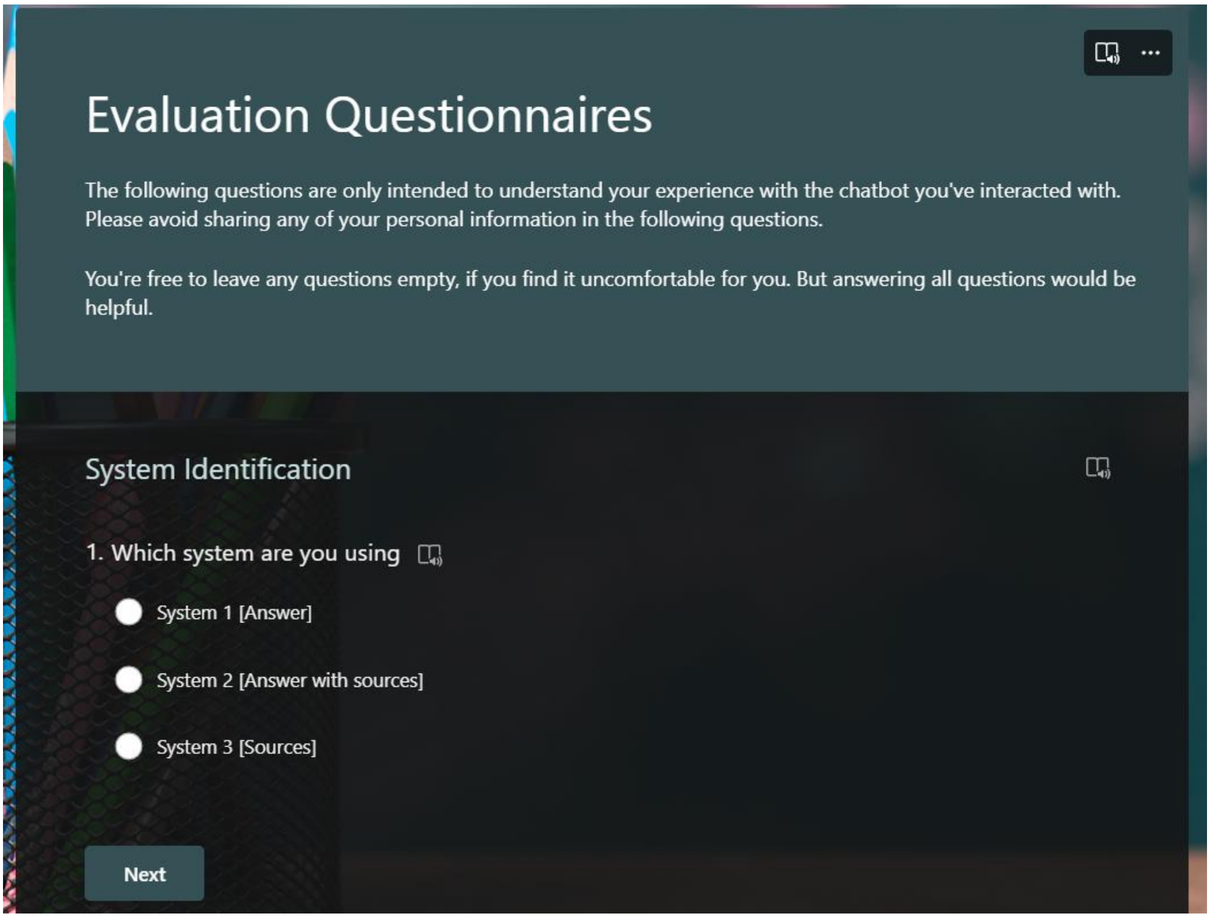
   Ask the following questions about the course:

   - Overview
   - Location
   - Entry requirements
   - Duration
   - How to apply

4. **Ask to talk about Economics MA (Hons):**

   Ask the following questions about the course:

   - Overview
   - Course contents
   - Program fees
   - Type of course delivery



Evaluation Questionnaires

The following questions are only intended to understand your experience with the chatbot you've interacted with. Please avoid sharing any of your personal information in the following questions.

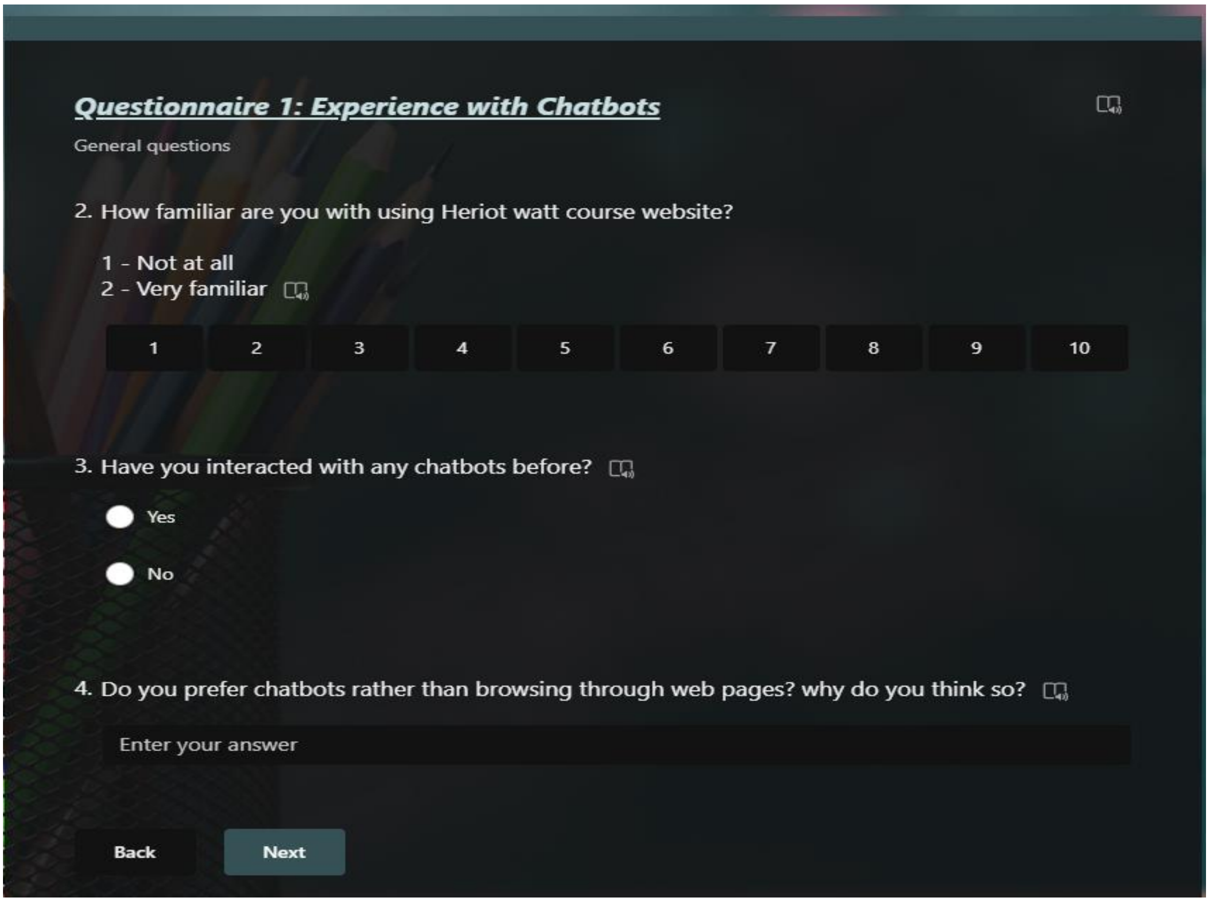
You're free to leave any questions empty, if you find it uncomfortable for you. But answering all questions would be helpful.

System Identification

1. Which system are you using

- System 1 [Answer]
- System 2 [Answer with sources]
- System 3 [Sources]

Next



Questionnaire 1: Experience with Chatbots

General questions

2. How familiar are you with using Heriot watt course website?

1 - Not at all
2 - Very familiar

1	2	3	4	5	6	7	8	9	10

3. Have you interacted with any chatbots before?

- Yes
- No

4. Do you prefer chatbots rather than browsing through web pages? why do you think so?

Enter your answer

Back Next

6. The chatbot provided very accurate answers for my questions

1- Strongly disagree
10 - Strongly agree

1	2	3	4	5	6	7	8	9	10

7. How satisfied are you with the information provided by the chatbot?

1- Not at all
10 - very satisfied

1	2	3	4	5	6	7	8	9	10

8. To what extent did you trust the answers provided by the chatbot?

1- Not at all
10 - completely

1	2	3	4	5	6	7	8	9	10

9. Did you feel the chatbot's responses were personalized to your needs and preferences?

- Yes
- No
- Not sure

10. How likely are you to recommend the chatbot to others for accessing university course information?

1 - Very unlikely
10 - Very likely

1	2	3	4	5	6	7	8	9	10

References:

1 Suta, P., Lan, X., Wu, B., Mongkolnam, P. and Chan, J.H., 2020. An overview of machine learning in chatbots. International Journal of Mechanical Engineering and Robotics Research, 9(4), pp.502-510.

2 Cerf, V., 1973. RFC0439: PARRY encounters the DOCTOR.

3 https://www.boost.ai/chatbot

4 Baidoo-Anu, D. and Owusu Ansah, L., 2023. Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning. Available at SSRN 4337484.

5 Swathilakshmi, V., 2021. Student suite+, a closed domain question answering system for educational domain. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), pp.3168-3172.

6 Tran, P., Nguyen, D., Tran, H.A., Nguyen, T. and Tran, T., 2023. Building a Closed-Domain Question Answering System for a Low-Resource Language. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(3), pp.1-14.

7 Zhou, X., Nurkowski, D., Menon, A., Akroyd, J., Mosbach, S. and Kraft, M., 2022. Question answering system for chemistry—A semantic agent extension. Digital Chemical Engineering, 3, p.100032.

8 Satapathy, S.C., Raju, K.S., Shyamala, K., Krishna, D.R. and Favorskaya, M.N. (2019) 'A Review on Different Question Answering System Approaches', in Advances in Decision Sciences, Image Processing, Security and Computer Vision. Switzerland: Springer International Publishing AG, pp. 579–586. doi:10.1007/978-3-030-24318-0_67.

9 DERİCİ, C. A. N. E. R., AYDIN, Y. İ. Ğ. İ. T., YENİALACA, Ç. İ. Ğ. D. E. M., AYDIN, N. İ. H. A. L. Y. A. Ğ. M. U. R., KARTAL, G. Ü. N. İ. Z. İ., ÖZGÜR, A. R. Z. U. C. A. N. and GÜNGÖR, T. U. N. G. A. (2018) "A closed-domain question answering framework using reliable resources to assist students," Natural Language Engineering. Cambridge University Press, 24(5), pp. 725–762. doi: 10.1017/S1351324918000141.

10 Khanbhai, M., Anyadi, P., Symons, J., Flott, K., Darzi, A. and Mayer, E., 2021. Applying natural language processing and machine learning techniques to patient experience feedback: a systematic review. BMJ Health & Care Informatics, 28(1).

11 Halper, F., 2017. Advanced analytics: Moving toward AI, machine learning, and natural language processing. TDWI Best Practices Report.

12 Quarteroni, S., 2018. Natural Language Processing for Industrial Applications. Spektrum, 41, p.105.

13 Lin, C.C., Huang, A.Y. and Yang, S.J., 2023. A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022). Sustainability, 15(5), p.4012.

14 Lee, H. and Jeong, O., 2023. A Knowledge-Grounded Task-Oriented Dialogue System with Hierarchical Structure for Enhancing Knowledge Selection. Sensors, 23(2), p.685.

15 Klaus, P. and Zaichkowsky, J. (2020) 'AI voice bots: a services marketing research agenda', The Journal of services marketing, 34(3), pp. 389–398. doi:10.1108/JSM-01-2019-0043.

16 Sonbol, R., Rebdawi, G. and Ghneim, N., 2022. The use of nlp-based text representation techniques to support requirement engineering tasks: A systematic mapping review. IEEE Access.

17 Sushil, M., Suster, S. and Daelemans, W., 2021, June. Are we there yet? Exploring clinical domain knowledge of BERT models. In Proceedings of the 20th Workshop on Biomedical Language Processing (pp. 41-53).

18 Zhang, M. and Li, J., 2021. A commentary of GPT-3 in MIT Technology Review 2021. Fundamental Research, 1(6), pp.831-833.

19 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

20 Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[21] *Bruch, S., Lucchese, C. and Nardini, F.M., 2023, January. Report on the 1st Workshop on Reaching Efficiency in Neural Information Retrieval (ReNeuIR 2022) at SIGIR 2022. In ACM SIGIR Forum (Vol. 56, No. 2, pp. 1-14). New York, NY, USA: ACM.*

22 Tunstall, L., Werra, L. van. and Wolf, T. (2022) Natural language processing with Transformers : Rev. ed. Sebastopol, Calif.: O'Reilly.

[23] *Caldarini, G. and Jaf, S., 2023. Recent Advances in Chatbot Algorithms, Techniques, and Technologies: DESIGNING CHATBOTS. Trends, Applications, and Challenges of Chatbot Technology, pp.245-273.*

[24] *Chan, C. and Li, F., 2023. Developing a natural language-based AI-chatbot for social work training: an illustrative case study. China Journal of Social Work, pp.1-16.*

[25] *Raj, V. and Phridviraj, M.S.B., 2023, March. A Generative Model Based Chatbot Using Recurrent Neural Networks. In Advanced Network Technologies and Intelligent Computing: Second International Conference, ANTIC 2022, Varanasi, India, December 22–24, 2022, Proceedings, Part II (pp. 379-392). Cham: Springer Nature Switzerland.*