UNIVERSITY OF REGINA
FACULTY OF GRADUATE STUDIES AND RESEARCH

DEPARTMENT OF COMPUTER SCIENCE

CS 820 PROJECT REPORT

HealthBotAssist: A Conversational AI in Healthcare

SUBY SINGH (SSZ389@UREGINA.CA)
VINEEL GANNU(VGC361@UREGINA.CA)

Abstract

Chatbots, a virtual assistant is the automated conversational application of technologies, such as Artificial Intelligence (AI). Users get engaged with chatbots by asking a query and receiving a response with zero intervention from a human. It is comprehensively being used in businesses such as providing customer support. AI can also help us progress to the future healthcare system. AI chatbot solutions in healthcare can have the ability to ease life for doctors, patients, nurses, and other hospital personnel in ways, such as preliminary symptoms examining, mass information circulation, emergency call, patients' regular follow up post-treatment, booking doctor's appointment, and so on. As a part of the CS820 project, we would like to propose one such preliminary AI assistant in healthcare system, named HealthBotAssist that will engage with users or patients to provide them preliminary assessment to symptoms that they observe regarding their health and guide them to understand their medical conditions accordingly, from the comfort of their homes. In such application of chatbots, finding a suitable recommendation is dependent on multiple constraints, such as severity of the symptoms. It concerns the realization of selecting an action and response sequence and needs to provide an intelligent and suitable response. Hence, this module typically concerns with the constraint satisfaction problem of AI. We implemented the proposed HealthBotAssist chatbot by applying AI techniques such as Natural Language Processing (NLP) and Machine learning (ML) to understand the intent of the conversation and the user's personal preferences. For our bot implementation, we observed forward sweep procedure is the best suit method for finding an optimal solution for the CP-nets because it constructs the most preferred outcome.

Keywords— Conversational Artificial Intelligence . Healthcare . Automated planning and scheduling . Preference selection.

Contents

1 Introduction

In today's fast-paced world, consumers expect easy access to information. As with other consumer sectors, healthcare consumers have similar expectations of their healthcare providers. One way to meet these consumer's expectations is to use a chatbot. A chatbot is a computer software that carries out conversations without any human intervention. Chatbots could be helpful during the time of pandemic for the health-care sector by providing guidance to the health condition seekers. It can also help health-care organizations by providing support for preliminary symptoms assessment and minimizing the cost of services by performing tasks like booking appointments, diagnosing diseases and many more.

In this project, we designed an AI assistant chatbot called HealthBotAssist. It does preliminary assessments of the symptoms observed by the users and guide them in learning about their medical conditions in the comfort of the user's own home. User is asked questions based upon their initial provided symptoms and bot tries to collect more details about user's health condition by establishing a conversation with them. The conversations are established based on extracting information about some constraints that should be considered before recommending any medical treatment. For example, medical background, duration of the symptoms, age, etc. Apart from this, the healthBotAssist has also been implemented to book appointment with a doctor. So, some other constraints such as preferred doctor, suitable date and time are also extracted from user's conversation with the bot. Constraint solving models based on preference are taken into consideration while designing the bot Assistant.

In many application domains, the decisions that can be made available to user is fixed or the space of possible actions at a particular steps is fixed. There would be some variable component where user provided preferences could be used to reach and automated decisions. Medical domain is the application of one such preference reasoning concept of AI. We have used this concept while designing the HealthBotAssist. The prototype for HealthBotAssist chatbot is shown in the figure 1. While working on the application of chatbot it is desirable to assess the preferences of users in a qualitative rather than quantitative way. For example, while booking an appointment with a doctor for the user, chatbot must know the preferred date and time which is suitable as per the user. Such preferences can be represented in qualitative preference ordering form rather than quantitative. This representation and adoption of qualitative form builds an important component of the automated decision tools.

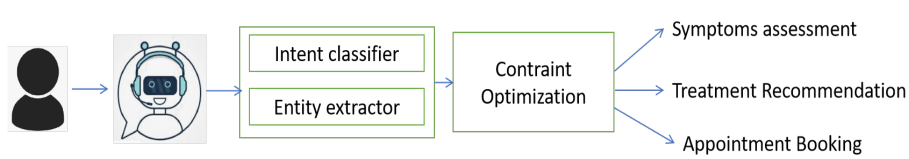

Figure 1: Prototype of HealthBotAssist

In case of treatment recommendation, use of the chatbot is expected to be satisfied by a solution that is identified by the bot and that satisfies all set of constraints. In appointment booking module, however users have preferences over the set of solutions, and user can expect a best satisfying solution, or atleast a good assignment with respect to his preference. For example, in case of booking an appointment for covid test, user have preferences for covid center and suitable time. Here, we try to find an optimal solution according to user's preferences and one can refer to this strategy as constraint optimization.

In this project implementation, we approach CP-nets to have graphical representation of user preferences. This graph can reflect the conditional dependence and independence of user preferences. We use branch-and-bound algorithm to obtain a set of Pareto optimal solutions that satisfy a set of constraints. We used this algorithm because it has an important anytime property. The set of solutions it generates is guaranteed to contain only Pareto-optimal solutions at any point of time. Saying this no solutions will also be retracted as new solutions, (i.e., no appointment booked for covid test can be considered as a solution in case the bot is not able to find a good assignment satisfying user preferences).

2 Background

In recent years, chatbots have come up as a hot topic in the new emerging technology that is being used by by numerous companies in various areas, such as automatic telephone answering system, help desk, e-commerce and so on. Recent advancement in messaging applications and AI technology have made chatbot's application worth to be focused on. It combines both AI and Information retrieval techniques. One of the examples of chatbot could be given by Sophia. It's a robot that was designed to provide assistance in mathematics by answering students' questions at Harvard. This concept was later applied to different context, for example 'Kommune Kari' in Norway. It's a chatbot that the municipality have it inbuilt on their web-pages. Kari answers to the questions like 'when will the garbage truck arrive?' or 'where can somebody find available jobs?". It's job is to provide user with the information related to such simple queries. so that they don't have to look into the 'massive information flow' (Schibevaag, 2017).

2.1 Existing work

There have been numerous advancements in artificial intelligence (AI), influencing how we interact and communicate with machines. Several industries have embraced chatbots, including business, health, e-commerce, and education. Businesses utilize chatbots because they reduce costs and are capable of handling multiple customers at once.

Eliza, a chatbot built by Joseph Weizenbaum at the Massachusetts Institute of Technology (MIT), is considered to be the first in the history of Computer Science. It makes use of pre-programmed responses to reproduce responses by recognizing key words and phrases from the input. It uses a method of pattern matching and substitution. Despite being mechanized, this process created the illusion that you were talking to somebody when you weren't.

Psychiatrist Kenneth Colby developed Parry in 1972 as an attempt to mimic the symptoms of schizophrenia. It is a program that uses natural language that mimics the way an individual thinks. Parry overruled the human interrogator's ability to differentiate it from a human, which made it the first bot to pass the Turing test.

A British programmer named Rollo Carpenter created a bot called Jabberwacky. the purpose of this bot is to "create realistic simulated human chat in an amusing and entertaining manner". This project's intent was to construct a machine that can pass the Turing Test and perform business tasks with artificial intelligence. Jabberwacky uses contextual pattern matching technique to simulate the chat.

2.2 Constraint Satisfaction Problem

Md. Ahsan Ayub et al.[1] have done an exhaustive study of Essential Constraint Satisfaction Problem Techniques based on N-Queens Problem where they formulated various real life problems as an instance of CSP that has large number of constraints. A fixed problem domain is considered where time vs value of N in N-Queens results have been derived for comparison. The study helps to instantiate a real-life CSP problem by taking the extended decisions.

Moni Sahu et al.[2] finds a solution for a Classical Crypt Arithmetic Problem (BASE+BAL L=GAMES). The proposed method enables us to easily calculate the solution for a problem, regardless of its nature and the method says that if the problem is larger, you can divide it into smaller pieces for better decisions. The method works efficiently for larger problems and is also more efficient in memory storage and time consuming. In comparison with the parallel genetic algorithm, the proposed method performed better.

A multi-agent method known as CSC (constraint satisfaction community) is presented by Raziyeh Moghaddas et al.[3] that solves constraint satisfaction problems having Alldiff constraints. The proposed algorithm has a linear time complexity for problems of larger size. By creating various communities and by interacting with each other, agents will try to improve their scores and the scores of the members of their communities. To test the performance of this algorithm, the N-Queens problem is used as a bench mark. It was tested with 4-queen up to 10^4-queen and the algorithm was able to find a legal solution in a reasonable time.

2.3 CP-net

Let us assume a set of variables $V = X_1, ..., X_n$ with finite domains $D(X_1), ..., D(X_n)$. A preference ranking is given by a decision maker over complete assignments on V. The assignments can be seen as possible outcomes of the decision maker's actions, which may be in the form of physical actions or, as is the case for configuration problems, merely choices of values for each variable. The set of all outcomes is denoted O. A preference ranking is a total preorder $\succeq$ over the set of outcomes: $o1 \succeq o2$ means that outcome o1 is equally as good as or preferred to o2 by the decision maker. We use $o1 \succ o2$ to denote the fact that outcome o1 is strictly preferred to o2 (i.e., $o1 \succeq o2$ and $o2 \not\succeq o1$), while $o1 \sim o2$ denotes that the decision maker is indifferent between o1 and o2 (i.e., $o1 \not\succeq o2$ and $o2 \not\succeq o1$). The preference ordering and relation are used interchangeably with ranking. [4]

Boutilier et al. (1999) introduced the CP-net as a way of representing qualitative preference relations compactly. It is a conditional preference network can be used to describe the qualitative conditional preferences of attributes. The Ceteris Paribus assumption captures the notion of purely qualitative preferential independence in CP-networks. They bear a superficial resemblance to Bayesian networks. When compared with the probabilistic relations in Bayes nets, the node relationships within a CP-net is weak. [4]

2.4 Constraint Optimization

Provided an acyclic CP-net and either an empty or partial assignment for its variables, finding an optimal decision or solution becomes straightforward. We can find an optimal solution in linear time order of the variables. There may also include hard constraints that may hinder in finding an optimal solution. In this case, our goal would be to find a feasible solution that is consistent with the hard constraints. In this we look for Pareto optimal outcome (that is for a given set of feasible solutions s, o is pareto optimal with respect to preference order $\succ$ iff there is no o' from the set S such that $o' \succ o$.

Our HealthBotAssist is implemented based on NLP (Natural Language Processing) for finding intent of the message. Upon finding the intent, constraints are solved and a preferred response is given to the user while finding an optimal solution for CP-nets.C

3 Implementation

HealthBotAssist project has been implemented on the open source Botpress Virtual Assistant platform that provides tools for developing, testing, and deploying AI-based conversational assistants. Our AI assistant assesses the symptoms of headache, cancer, and covid. It guides a user to learn about their medical conditions and provides a recommendation based on the symptoms and its severity. In case of each disease assessment, these symptoms play the role of constraints and can be solved to assign a recommendation. Bot finds a suitable recommendation that satisfies all the constraints in terms of symptoms. Apart from this, bot also determines if user needs to consult a doctor and helps user in booking an appointment with the clinical center. Appointment booking problem can be represented using a graphical method of CP-net as shown in Figures 5 and 6.

3.1 Symptoms Assessment and Recommendation based on Constraint Satisfaction Problem

Each of the following modules corresponds to the constraint satisfaction problem. Variables represent a set of questions and domains represent possible answers for a given question. Constraints represent the user's answers and the CSP solution represents the standard of medical care recommended to the user. We calculate severity of the overall symptoms by assigning every symptom a weight between 1 and 10 for a particular disease. Based on the final calculation of the severity, one of the available recommendation (outcome) is selected.

3.1.1 Headache

While recommending a treatment for headache, bot first solves a constraint of injury. If user's head is hurting due to an injury then bot immediately suggests to visit a doctor and helps in booking an appointment for the same. Otherwise, bot tries to determine whether the user normal headache or migraine. Following are the constraints for determining migraine: Duration of the headache, vomiting or nausea sensation, sensitive to light, duration for feeling sensitive to light

If user specifies that he just has thirst, dry skin, muscle cramps, throbbing pain then user might have normal headache due to dehydration. Later, bot assigns one of the following listed outcome (recommendation) based on the user's input and provides it to the user:

- Recommendation 1: Headache due to dehydration. (recommendation: massage, stay hydrated drink water Electrolyte drinks, take medicines like Pedialyte and Powerade OTC pain relievers (Advil, aspirin, and Tylenol))
- Recommendation 2: Minor migraine issue. (recommendation: stay in a dark or quiet Room, perform breathing exercises, stay hydrated, take proper sleep, exercise or do yoga daily)
- Recommendation 3: Migraine. (recommendation: may take Imitrex to treat up to four migraine or cluster headaches a month or sumatriptan, stay in a quiet and dark room, relaxation and yoga)

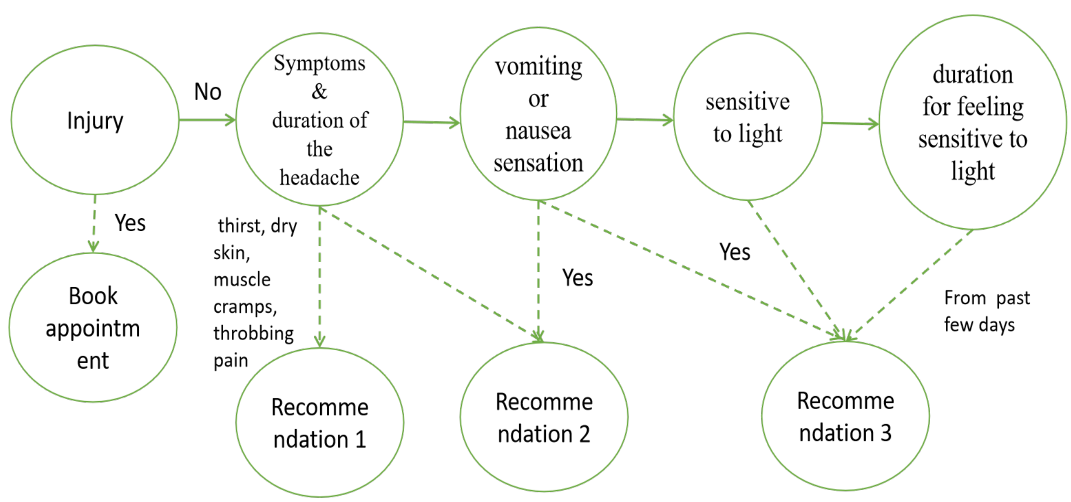

Figure 2: Headache Module

3.1.2 Cancer

AI Assistant executes the cancer module if the intent of the user is to assess early cancer symptoms. The AI Assistant asks questions related to the symptoms of bleeding, finger and toe nails, fatigue and weakness, skin changes and swallowing issues. The AI Assistant uses these symptoms to determine whether the user is suffering from cancer. The user may be given a medication after the preliminary assessment, or a second preliminary assessment may be made to suggest a better medical treatment based on their input. The second part of the assessment involves body monitoring, in which the user is asked to observe a few signs and notice any changes. Using these assessments, the AI Assistant better understands the condition of the user. The AI Assistant also schedules an appointment for the user.

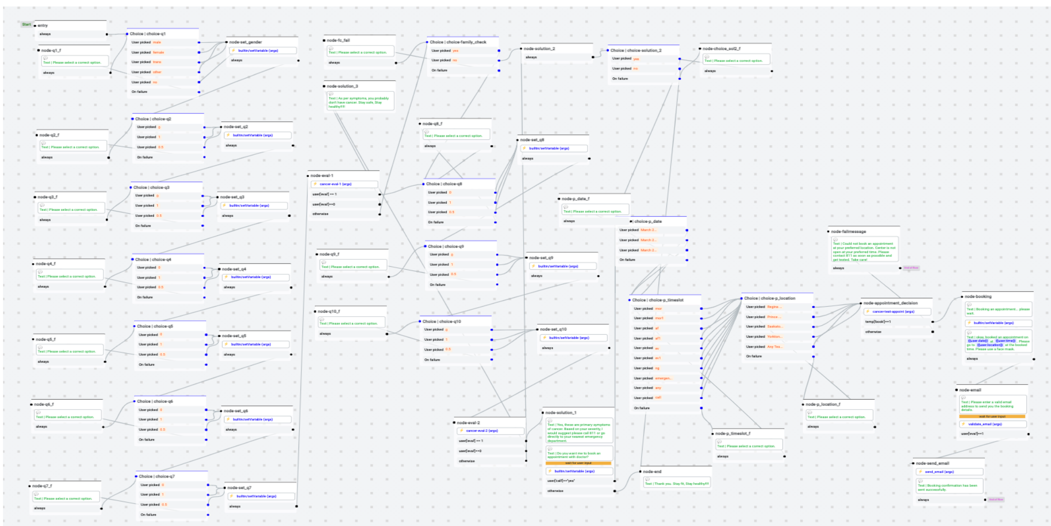

Figure 3: Cancer Module

3.1.3 Covid

While assessing symptoms for covid, bot responses in four outcome based on constraints optimized. As per bot implementation, the highest priority is given to following symptoms (constraint for getting tested for covid) : difficulty in breathing, pain in chest, feeling confused, losing consciousness. If user's response satisfies this constraint, user is asked to get tested from one of the available covid test center (CP-net in Figure 6 shows the constraints over booking an appointment for covid test).

Other symptoms details such as fever, cold, headache, sore throat and constraints such as any undergoing treatment, historical health condition, contact with a covid positive person, how long for contact, and age are also considered while determining one of the following outcomes:

- Recommendation 1: Stay home Monitor your health Face coverings and masks Take preliminary medicines for the symptoms
- Recommendation 2: Get tested Except for testing, stay home Tell people you have been in close physical contact with Monitor your health
- Recommendation 3: no covid, but take Tylenol for fever or headache!!

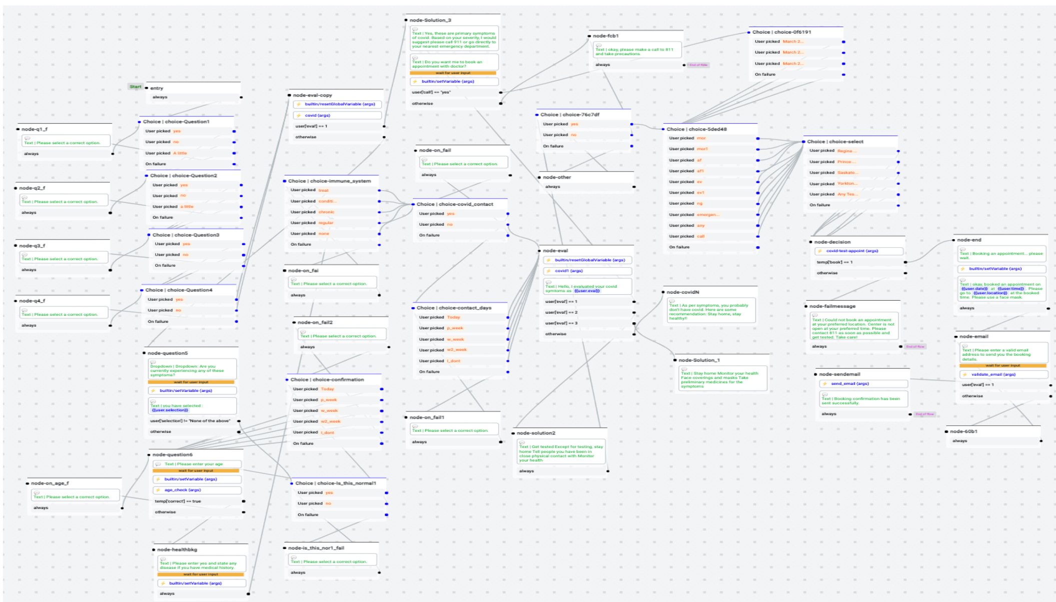

Figure 4: Covid Module

3.2 Appointment Booking : Graph representation using CP-net

Along with the medical treatment recommendation, the AI Assistant can book an appointment with a doctor. It uses constraint optimization where a solution to a constraint satisfaction problem is found by optimizing the user preferences. The appointment date, appointment time and the appointment location are considered as the constraints. The user's qualitative preferences are represented using the CP-net that addresses the optimization problem.

In Figure 5 according to the CP-net N for booking appointment in case of headache, we have three constraint variables, i.e., V = {Date, time, clinic}. If the user wants bot to book an appoint on his behalf, the bot first takes preference for time and date. Depending on User preference for time and date, one of the clinic is selected. Here, time could be starting from morning 8:00 AM to 10:00 PM (each 2 hours slot) and the date entered by User is used to decide if its the weekend or weekday.

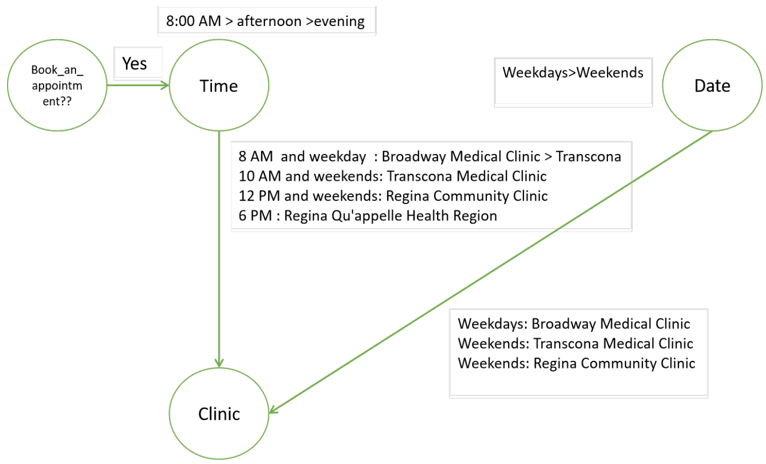

Figure 5: CP-net *N* for appointment booking (Headache)

Depending on User preference for time and date, one of the available clinics is determined by the bot. For example, if user wants to see a doctor on 30-march-2019(weekday) at 8:00 AM in the morning, Broadway Medical Clinic will be assigned to the user's preferred query.

In Figure 6, as per the CP-net *P* for appointment booking due to Covid systems, we have four constraint variables, i.e., V = {Covid Testing Center, Date, time, test mode}. User has highest preference over selecting a Covid Testing center and date. Depending on the selected Covid center and a date (date could decide today, tomorrow or later) the bot determines a preferred time slots based on the CPT provided for time variable. Once time slot is determined, test mode is also finalized based on the CPT provided at test mode variable. Here, in this example we have shown that if user prefers Regina over other covid testing center and provides today's date then a most preferably a afternoon shift is selected (because Regina covid center starts at 12:00 PM). Later, depending on the selected time, test mode is also selected by the bot. If an afternoon slot is assigned by bot to time variable then drive-thru is preferred over in-person testing.

3.3 Booking Optimization

One of the principle factors of CP-net is that we find the best outcome that preferences ranking and satisfies N for a given CP-net *N*. CP-nets make this task simple. Intuitively, we traverse from the parent constraints or ancestors to descendants assigning each variable with most preferred value while sweeping throughout the network. This procedure is known as forward sweep mechanism.

Given the acyclic CP-net *N* and *P* for appointment booking in Figure 5 and 6 respectively, we can determine the best suitable booking among those preferences that satisfy the CP-net *N* and *P*. Query *N* and *P* are called the Outcome optimization queries. Simply following forward sweep procedure, we sweep through the top node variables (time and date) to bottom node variable (clinic) in case of CP-net *N* and variables(covid test center and date) to time to test mode in case of CP-net

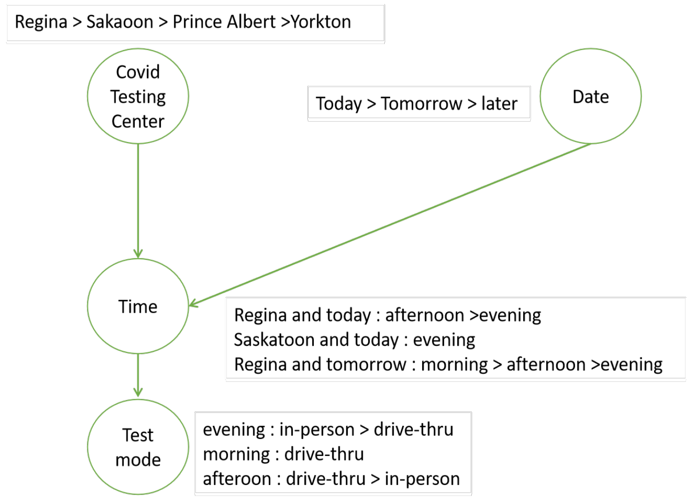

Figure 6: CP-net P for appointment booking (Covid)

P, setting each variable according to most preferred value. Ideally the network finds a unique best outcome following forward sweep procedure and giving certain observed evidence.

4 Conclusion

In this project, we implemented a chatbot, HealthBotAssist that performs preliminary assessment about the users' health conditions and helps them learn about their medical condition. Questions are posed based on the user's initial symptoms and the bot attempts to gather more details by conversing with them. Contextual information is derived from the conversations to determine the boundaries imposed on recommending any medical treatment. Also, healthBotAssist can be used to book appointments with doctors. Additionally, from the user's conversation with the bot, other constraints may also be extracted, such as preferred doctor and date and time. When designing the Assistant bot, preference-driven constraint solving models are taken into consideration.

5 Acknowledgements

We are exceptionally grateful to Dr.Malek Mouhoub for his guidance and continuous supervision as well as for providing us with necessary information regarding our project, helping us finish the project, and providing support in achieving our goals.

6 References

[1] M. A. Ayub, K. A. Kalpoma, H. T. Proma, S. M. Kabir and R. I. H. Chowdhury, “Exhaustive study of essential constraint satisfaction problem techniques based on N-Queens problem,” 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2017, pp. 1-6, doi: 10.1109/ICCITECHN.2017.8281850.

[2] M. Sahu, A. V. Singh and S. K. Khatri, “A Classical Constraint Satisfaction Problem and its Solution using Artificial Intelligence,” 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 429-433, doi: 10.1109/AICAI.2019.8701325.

[3] R. Moghaddas, E. Mansoori and A. Hamzeh, “A new multi-agent algorithm for solving constraint satisfaction problems,” The 5th Conference on Information and Knowledge Technology, Shiraz, Iran, 2013, pp. 197-201, doi: 10.1109/IKT.2013.6620064.

[4] Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H. and Poole, D. (2004), Preference-Based Constrained Optimization with CP-Nets. Computational Intelligence, 20: 137-157. https://doi.org/10.1111/j.0824-7935.2004.00234.x

[5] https://www.spottingcancer.org/