A Hybrid Approach to Develop and Integrate Chatbot in Health Informatics Systems

Abhijat Chaturvedi, Siddharth Srivastava, Astha Rai and A. S. Cheema
Centre for Development of Advanced Computing, Noida, India

Keywords: Chatbot, Health Informatics Systems.

Abstract: In this paper, we develop a chatbot that seeks free-form natural language queries by its users for blood and related services such as list of blood banks, live blood stock, blood donation camps etc. with one or more parameters as search criteria. The queries can be both Frequently Asked Questions (FAQs) and data driven including location based services. The uniqueness of this chatbot lies in the fact that its architecture provides it flexibility to evolve to encompass more domains and services without having any impact on existing services. Moreover, with approximate keyword initialization, the proposed chatbot can smartly infer from incomplete or incorrect queries by the users as well as has the ability to learn abbreviations. The bot achieves this by leveraging state-of-the-art deep learning and natural language understanding algorithms at the back-end. Specifically, this bot uses a hierarchical approach for parsing queries. At first level, it parses the query into intents i.e. FAQ or data driven. If the classified intent is a FAQ, chatbot to respond while, if it is amongst many of the citizen centric queries, it drills down through the query to identify the entities such as city etc. along with the type of the service and returns the users with the required details.

1 INTRODUCTION

As the reachability of various health informatics solutions are increasing, the usability and effectiveness of such systems have become important. While user interfaces and workflow play an important role in acceptance of such systems, but letting the users query the system with natural text as input and giving them the appropriate output such as data, help text, excerpts from user manuals, filtering their query for more inputs, is becoming the next level of expertise required in health informatics solutions. The chatbots are a natural solution to this problem (Chung and Park, 2019). Chatbots brings the ease of use along with 24X7 service at a fraction of what a human employee would require. This enhances the operational efficiency at a reduced cost. As argued by Petter Bae Brandtzaeg in his study(Brandtzaeg and Følstad, 2017) people prefer chatbots because of productivity. Chatbots in health care provides a variety of solutions from symptoms checking (Divya et al., 2018) to getting basic health information (James and Vales, 2009). The AI provides the edge to the chatbots over any other technology, because they can learn and grow over time. This saves the time and money of both patients and health care provider. Thereby, creating a smooth health care experience.

However, most of the chatbot solutions work on static data or pre-defined workflows which may cater to specific requirements. Therefore, in this work, we propose a modular chatbot framework for requirements suiting to health informatics solution. To demonstrate the effectiveness of the framework, we take a "Blood Bank Management System" as a use case. Specifically, the considered blood bank management system has two modes of usage. The first is via a citizen centric portal where details on blood banks, blood stock, information etc. are given. The second is the set of users using the application for day to day use such as blood bank staff. Therefore, we implement the proposed framework for addressing queries from citizens via a nationwide public web portal as well as the users of a blood bank management system.

Figure 1 shows the workflow of queries from a citizen centric portal for a blood bank within the proposed architecture. It is important to note that, the queries from both sets of users (citizens, application users) are different. While we can limit the scope of queries from internal users of the health informatics solutions such as blood bank staff in the considered use case, it is extremely difficult and impractical to control or limit the queries from citizens on a public

Chaturvedi, A., Srivastava, S., Rai, A. and Cheema, A.
A Hybrid Approach to Develop and Integrate Chatbot in Health Informatics Systems.
DOI: 10.5220/0009188007740781
In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 5: HEALTHINF*, pages 774-781
ISBN: 978-989-758-398-8; ISSN: 2184-4305

portal.

In view of the above, the contributions of this paper are:

1. A generic chatbot framework for health informatics solutions and implementation in a nationwide blood bank management system.
2. Formalizing common problems and their solutions while implementing chatbots.

The rest of the paper is organized as follows, in Section 2 we discuss and compare various types of chatbots. In Section 3, we formally define the problem, the challenges and relevant works. Section 4 describes the proposed architecture while Section 5 discusses the results. In Section 6 we discuss possible extensions of the proposed architecture followed by conclusions in Section 7.

2 TYPES OF CHATBOTS

On the basis of workflow, chatbots can be classified into three categories as follows

- **Menu based Chatbots:** Menu based chatbots work on the principle of decision trees. The hierarchies are used to get the next stage of input from the user and thereby classify the next best action based on the input received from the user. These chatbots full fill the basic search of the queries but the advance search with multiple variables can not be handled from such type of chatbots. Also, the menu based chatbots fall short on the number of clicks required for getting the desired information from the bot. It requires maximum number of flows to get the desired information among the three types of chatbots discussed in this paper.
- **Keyword based Chatbots:** These types of chatbots works on the identification of keywords. Once the input is received from the user, it tries to map with the existing set of keywords and once a match is found the corresponding logic is executed. This type of chatbots gives an illusion of understanding the user. However, it just identifies the keywords in the user's statement. Keyword based chatbots fails when there is multiplicity in the keywords and one keyword falls under different intents.

  Like in the case of blood stock search and blood donation camp search. In both the intents, keywords are same such as *blood* and *search*. Similarly, the FAQ's about blood donation also have many common keywords.
- **Contextual based Chatbots:** Contextual chatbots understands the context of the conversation and therefore are the most advanced form of chatbots. These bots utilize Machine Learning (ML) and Artificial Intelligence (AI) to remember conversations with specific users to learn and grow over time(Xiaojiang, 2014). These chatbots fills the slots once the information is available and then uses the same slots to understands the context of the conversation. This helps the user because the bot understands the history and flow of the conversation and remembers the information it has received from the user.

  A common example of flow of the contextual chatbot is when user asks the availability of B positive blood in Delhi. The bot replies the stock list of blood banks. Then the user inputs *and in mumbai?*. The bot understands that the user is asking for availability of B positive blood.

3 BACKGROUND

3.1 Problem

While the paper aims at developing chatbots for health informatics solutions and the proposed architecture is applicable to development of generic architectures, for clarity of the readers, we limit the scope of the discussion for next two sections w.r.t. a blood bank management system. Blood search, traditionally was done over the web in the drop down select type method, which is convenient but not enough as typing a random query over the app. The same goes with the frequently asked questions. Users do not want to read all the question, if they want to know the answer of just one. Therefore, to reduce the time, effort and ease the complexity for users, a system is needed which can be interactive and provide relevant information by understanding the language of the user, who may or may not be accustomed to the system. Therefore, we propose to use chatbot for interacting with the users and solving their queries for getting information or providing them help while using the system.

3.2 Challenges

One of the biggest challenge in the process of creating the bot was the understanding of the user's intention in the different sounding statements. For instance, the statements, *"Look for blood in Delhi"* and *"I am looking blood banks in Delhi, India"*, may have different selection of words, but the intention of the

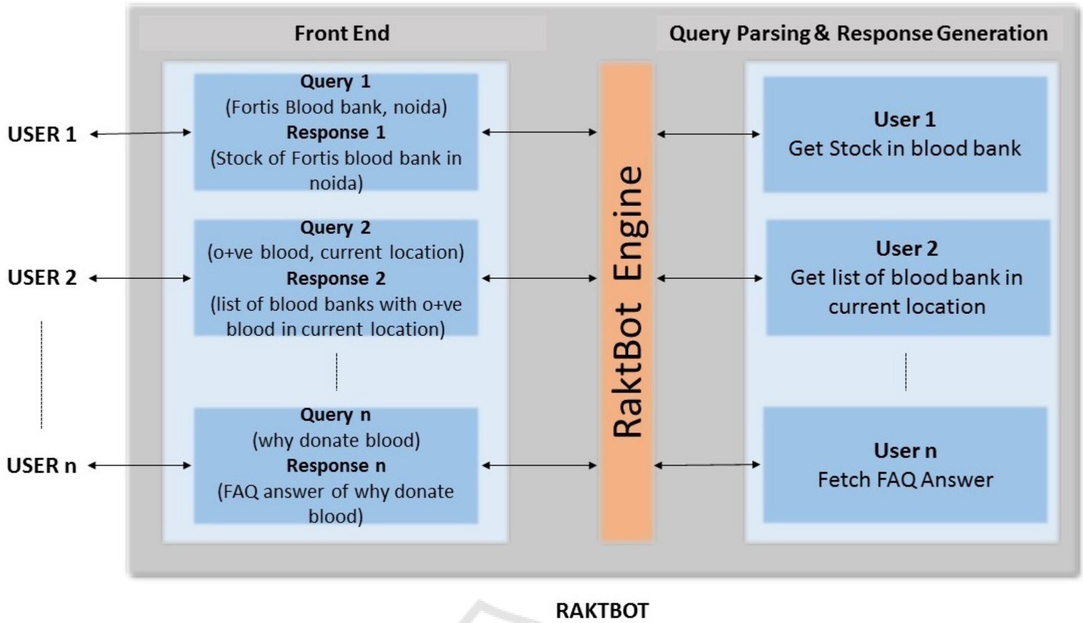

Figure 1: Workflow of the proposed framework.

user in both these sentences is the same. In both the cases the user wants to search the blood for donation in Delhi.

The correct mapping of the questions in the FAQ section was another major challenge. In practice, in any application, initially the FAQs are limited and they grow as the application is developed over the years and also with the number of users. Therefore, with limited amount of questions in the beginning, it is difficult to correctly relate the user's query to the nearest question available in the dataset. For example, when the questions in the dataset can be similar ("Why should I donate blood?", "When to donate blood?") and the input from users can be limited ("Why donate blood?"), the algorithms usually provide incorrect output as most of the sentences in the dataset contain the word blood and when, why ("When to donate blood instead of plasma?").

3.3 Relevant Works

There has been no attempt at making a blood donation search chatbot. Although there are many FAQ chatbots as well as many chatbots are tailored for healthcare industry, most of them are for disease prediction like Mandy (Ni et al., 2017) and Quro (Ghosh et al., 2018) or act as a counselling service (Lee et al., 2017). The majority of these FAQ chatbots deploy a bag-of-words model to predict the answer of the question entered by the user. This technique, is however, not always effective especially in the cases where the length of the questions is comparatively large.

3.4 Motivation

Ability to get information on availability of is a task of utmost urgency. We wanted to provide a way that is fast, easy to use and reliable when it comes to the question of life and death. Hence we choose to create a chatbot because that is the most easy way for a non technical person to search for blood when required.

Also, every business has its own challenges and requirements, it is not possible to fit one solution to every problem. Therefore, we wanted to provide a solution for the better engagement of the users on the business, through chatbots. Considering the majority of the people who do not understand the complex of higher machine learning, artificial intelligence and natural language processing, we created a modular approach to create a chatbot, which can be used as-is or extended by adding, removing or modifying the existing modules.

4 METHODOLOGY

The chatbot can is made up of two parts, one being the *FAQ* section which handles the FAQs and the other being the *custom* section for all the other queries of the user. In all total this bot handles 13 different intents. The architecture of the bot is displayed in Figure 2.

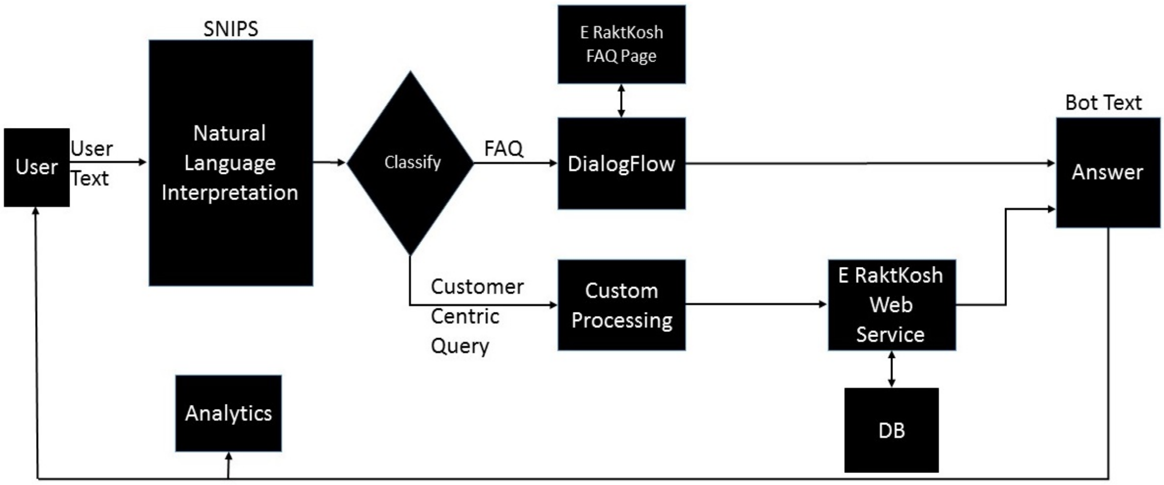

Figure 2: Architecture.

4.1 Intent Parsing

The first step that is made in the process of chatbot providing an answer to the user's query, once the user's input is captured is the intent classification or intent identification. For this, Snips NLU is deployed. Snips reads the text and based on it's Deterministic Intent Parser(DIP) and Probabilistic Intent Parser(PIP) classifies the intent and the corresponding entities as explained by Coucke et al. in 2018 (Coucke et al., 2018). The step is vital since the output of this step is the intent or the expectation of the user from the bot. This output is the input to the next step in the process.

The Snips algorithm reads the text entered and tries to map the statement with the statements provided in the training set. This is done by the component called Deterministic Intent Parser(DIP). This is swift and accurate segment of the Snips algorithm. However, in most cases it is not possible to get the exact statement as provided during the training phase. Therefore, the second segment Probabilistic Intent Parser(PIP) is put into place. If the DIP fails to predict the intent and entities PIP is initiated and on the basis of probability model, it identifies the intent and entities of the user's statement. See this structure in fig 2.

Once the intent is classified, next comes the task of generating the response corresponding to the user's query. This is done through a series of custom processing, business logic and api calls.

4.2 FAQ Section

To overcome the limitation of scarcity of data available to train, we used two fold mechanism. One is to generate similar sounding questions from all the questions in the dataset. This helped a lot in identifying the correct question and thereby correct corresponding answer that the user wants. Second was to put the bot to use by actual users. We therefore presented some different users who used and queried the bot just like an actual users. Each of them used the bot in all intents and all use cases. Further processing the FAQ's multiple options were available, including external libraries and internal machine learning model. In order to keep it simple and easy to use, we prefer pre-build libraries rather than creating a ML model from scratch. For this, many options were available (Janarthanam, 2017), including DialogFlow and MS Azure.

As studied by Canonico et al. in 2018 (Canonico and De Russis, 2018), DialogFlow is the best available cloud based chatbot engine. It has high usability, pre-build entities and pre-build intents. Also, it has Default Fallback intent, which comes handy when an unexpected statement is encountered.

The setup of DialogFlow started with the creation of a Google account and creating an agent that will process the information and produce the result. Once done, the FAQ's has to be uploaded in the csv format. The DialogFlow, automatically learns the intents and entities and trains the agent.

4.3 Custom Processing

For the search of blood along with the quantity and camp search, the custom processing section is provided. This section basically handles the queries of the blood search. Once the intent is classified the corresponding code gets invoked and the API calls are made to the e-Rakt Kosh Web Services.

The user's query contains the intent and the enti-

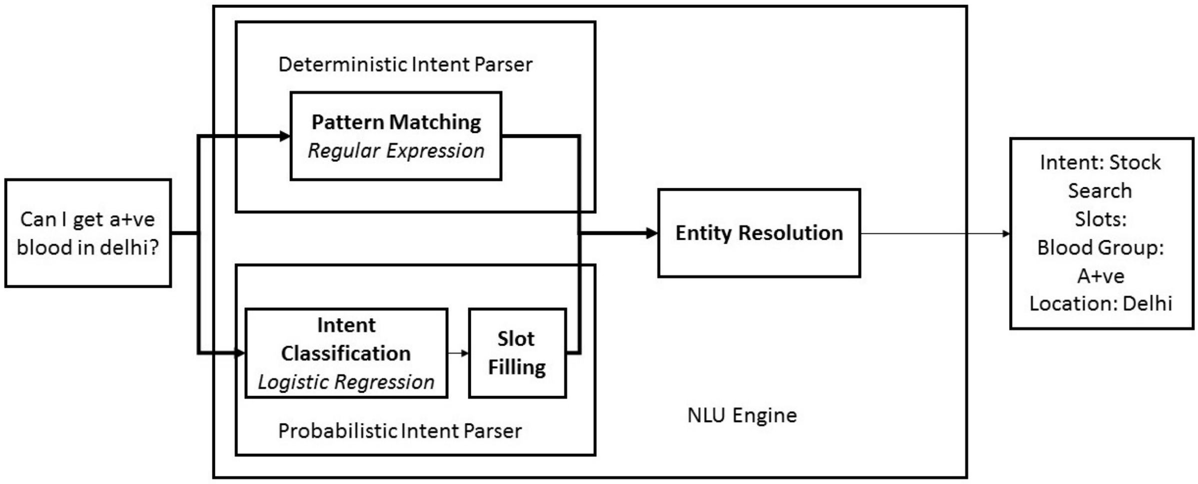

Figure 3: Chatbot framework based on Snips NLU architecture.

ties for the blood search. These intent and entities are identified by the Snips algorithm and upon identification the API of the e-Rakt Kosh web services are invoked, the API returns the data in json format, which is again parsed by the custom processing segment in order to present the desired data in the desired format.

Intents like mobile apps for download and help are also included for the better user aid. Quantity bound blood search is an important part where most user engagement is expected. Intent to contact the service provider is also put into place so as to give the user a smooth experience while chatting. This way users need to put a single statement for getting the quantity of blood available in a particular blood bank.

4.4 Analytics

MD Mulvenna et al. presented a way for analysing the chat logs (Mulvenna et al.,). For the sake of future development and to understand the usage pattern of the users, we have deployed the analytic mechanism for the bot. Through this we want to capture the insights such as the most searched question or which geographical area is using the bot most. This will help us in creating a better user experiences by enhancing the search and service accordingly.

The chat logs are captured in the SQL database, from where it can be accessed for the analysis and usage pattern understanding. Actionable analytics are of immense help in expanding the services and user experience in any business. The bot usage insights will result in the better and smooth user experience, which in turn enhance the user engagement.

4.5 Workflow

For instance the user enter the query, *"Look for ab+ve blood in delhi"*. The very first step in the process will be the parsing of the sentence in the Snips algorithm which will classify the intent of the user, which in this case is, Stock Search. The entities will be identified as "A+ve" and "Delhi". As they both are necessary for the fetching of the list of blood banks.

Once the intent is identified, it will lead to the corresponding segment to be initiated to generate the corresponding answer. Here, it is important to note that chatbot handles only one intent at a time. This allows the bot to be simple and light weight.

Since the intent here is Stock Search, therefore the bot will try to get the answer for that. The blood group is identified as "A+ve" and the city as "Delhi". Now, the bot will get the geocodes of the said location from Google Maps API. The geocodes are then sent to the e-Rakt Kosh web service, which returns the blood banks in the area.

Similarly, in case of fetching the list of blood banks along with the minimum required quantity of blood. The work-flow remains the same except after fetching the list, the bot filters out the blood banks which have less than required quantity of blood units. See workflow in fig 3.

On the similar lines, if the statement is like *"how much o+ve blood is avaiable in fortis hospital , noida"*, the entities that will be extracted will be "Fortis hospital" as the name of the blood bank, "O+ve" as blood group and "noida" as city. However the intent will be count search instead of simply stock search.

Some other less used intents are also put in places for one stop solution for the users. These includes, a help intent which guides the user to use the bot in

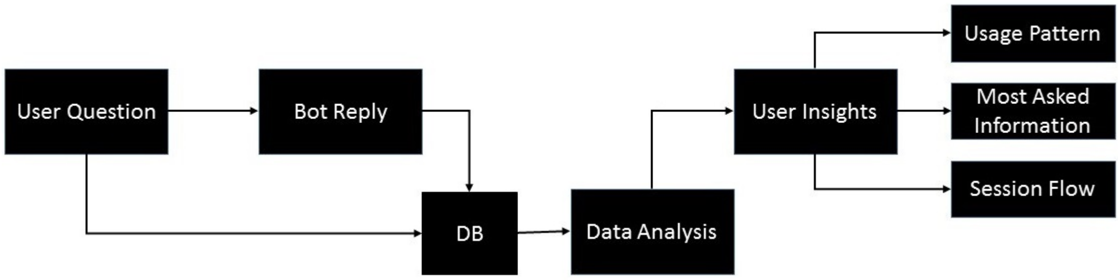

Figure 4: Workflow.

easy steps. Notification intent is also present to fetch any latest notification about the blood bank portal.

4.6 Chat Logs

Every interaction with the bot is logged in the e-Rakt Kosh database. This data is used for analysis of the user engagement and enhancing the bot's quality. The questions asked by the user are directly put into tables. Corresponding to which intent as classified by the bot is put in the table along with the response from the bot. Once the chat is started the logging process begins.

4.7 Integration

Once developed the bot needs to be integrated with the erkatkosh website. This was the biggest challenge of the project. The website is developed in java and the bot is developed in python. Integration of the both causes technical glitches and inefficiency.

Two clear options were available for the integration of the code. One was to compile the python code in java and generate a java byte code. This is however not a suitable mechanism because of its resource intensiveness. The other method was to use api's. This way it will be easy to maintain and less resource intensive. The most common to use python server is flask(Vogel et al., 2017)(Lokhande et al., 2015) which hosts the application. Therefore we had to create an instance of flask and use it's api's to get the response to the main page in e-Rakt Kosh website.

At the front end, there is the html, javascript and css page for e-Rakt Kosh website with a popup of the chatbot. This captures the users's statement in the textarea. Following it is the api call that sends this statement to the flask application running on another port on the same server through post method. The flask application is the bot application itself. It receives the input processes it through business logic and machine learning algorithm it has and sends back

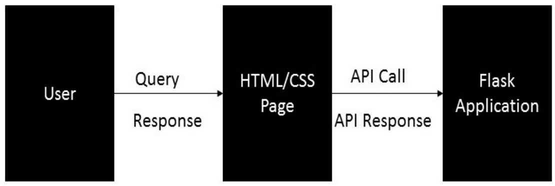

Figure 5: Flask API Flow.

the generated response. This structure is shown in figure 4. Flask API structure is shown in figure 5.

The simple and easy to use features are the key components of the bot. This is to ensure that users with all technical literacy are able to use the bot. Minimum button layout is preferred so as to clear the clutter and give the user a smooth experience.

5 RESULTS

In 1950, Alan Turing proposes the Turing test for the intelligent machines (Turing, 1950). Much later in 2016, Zhou Yu et al. proposes the crowd source method of evaluating the chatbot (Yu et al., 2016). They propose a method of taking user feedback on the chatbot usage. They argued that with modern chatbots, which are much more capable and there is no end to the chat(in general chatbots) the Turing test is not enough for performance evaluation. Marking the chat as satisfactory or not, and whether the user is willing to use the service again, provides a mechanism to evaluate the chatbot.

We have used the first method in our result fixation. A dataset of random 206 questions from all intents is used along with some questions that does not fall in any intent. Such a dataset is put to test the intent classification of the bot. Out of the 206 random questions, the bot correctly answered 179 questions, while 26 questions were not identified by the bot. This actually provides a near 87 percent accuracy.

However merely intent classification of the user's statement is not enough for the bot. Generating the correct statement for response is vitally important. In order to check the correct response generation, the same dataset is used to test the accuracy of the bot. Out of 206 questions, the bot accurately answered 175 question working at an approx accuracy of 85 percent.

The dataset is entered in the database and a script is used to fetch the question from this testing dataset and putting it into the bat and finally saving the bot's response back in the database. This created a table with a set of random question and answers on the basis of the chat session.

The second technique proposed by Zhou Yu et al. is based on the capturing of the user feedback. This evaluation technique is however, more subjective and user centric, therefore the results vary to the great extent. They propose a system where the users are able to rate the bot after a set of interaction with the chatbot.

Table 1: Intent Classification and Response generation results.

Results			
Measure	Correctly Identified	Incorrectly Identified	Accuracy
Intent Classification	179	27	86.89
Response Generation	175	31	84.95

6 GUIDELINES AND FUTURE DIRECTIONS

In this section, we provide a few suggestions for replicating and extending our work which will be of importance for complete integration of chatbots in health informatics systems.

- **Analysis:** The formal analysis of chatbot solutions in an indistrial setting is important but is given little attention. We recommend to setup an R based analytic tool for the proper analysis of the chat logs for both exploratory data analysis and explanatory data analysis. This will provide the business intelligence tools and graphs that are easy to present and read information from. The analysis also reveals the usage patterns and most queried statements. This type of data gathering is important for the better customization of the product. It will also be helpful in identifying the gap in the supply and demand of the blood in various parts of the country.
- **Voice based Chat:** The chatbots can be extended with TTS(Text to Speech) and STT(Speech to Text), making it further easy for anyone to use it. The voice capturing can be of immense help when being used in mobile devices, thereby tapping a huge market. There has been some of the works in this area as well. Works of Tsiao et al(Tsiao et al., 2007) and Emerick et al(Emerick et al., 2015) notable in this area. The voice based search combining with the regional language support will open this product to virtually everyone holding a smartphone. In the context of this work, this will let users search for blood in emergency situations, irrespective of their technical literacy and proficiency.
- **Language:** Efforts should be made to include regional languages along with English, so as to enable every person to use this service, thereby removing the language barrier. Vīra et al (Vīra et al., 2014) demonstrated this with Spanish, French, and Russian apart from English.
- **Conversational Responses:** The chatbot responds to the user messages if it is well trained on the possible flow of the conversation. This gives the user an illusion that the bot is actually talking to him. However, the bot is just following the pre-trained story line with different branches as defined by the developer team.

  Other libraries such as Rasa NLU and Rasa Core provides good enough structure to create a conversational chatbot(Bocklisch et al., 2017), that keeps the track of the conversation between the user and the bot and provides feature rich experience to the user. While we found that Rasa is more flexible, but it is slightly complicated (dependencies and required data) for quickly getting a system up and running as compared to Snips.

7 CONCLUSION

We have elaborated our proposed system in detail. The proposed system will bring the flexibility and ease in the blood search at the national level. With the use of Natural Language Processing, users can look for blood in query format rather than looking up on the web page. This saves time and effort which is vital in cases of emergencies. The future plans also includes the search in multiple languages which will make the system more accessible for other languages as well. The system will be the first point of contact between a potential donor and the e-Rakt Kosh application. It will cater the FAQ's to the donor and clear the doubts regarding blood donation.

We have explained its integration with the existing e-Rakt Kosh application and the challenges that were faced including the different language and environment

REFERENCES

Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Brandtzaeg, P. B. and Følstad, A. (2017). Why people use chatbots. In *International Conference on Internet Science*, pages 377–392. Springer.

Canonico, M. and De Russis, L. (2018). A comparison and critique of natural language understanding tools. *CLOUD COMPUTING 2018*, page 120.

Chung, K. and Park, R. C. (2019). Chatbot-based heathcare service with a knowledge base for cloud computing. *Cluster Computing*, 22(1):1925–1937.

Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., et al. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Divya, S., Indumathi, V., Ishwarya, S., Priyasankari, M., and Devi, K. (2018). A self-diagnosis medical chatbot using artificial intelligence. *Journal of Web Development and Web Designing*, 3(1).

Emerick, C. T., Logan, J. R., Bates, R. A., and Wahl, J. (2015). Voice assistant system. US Patent 9,171,543.

Ghosh, S., Bhatia, S., and Bhatia, A. (2018). Quro: Facilitating user symptom check using a personalised chatbot-oriented dialogue system. *Studies in health technology and informatics*, 252:51–56.

James, S. and Vales, I. (2009). Personal healthcare assistant/companion in virtual world. In *AAAI Fall Symposium Series*.

Janarthanam, S. (2017). Hands-on chatbots and conversational ui development: Build chatbots and voice user interfaces with chatfuel, dialogflow, microsoft bot framework, twilio, and alexa skills.

Lee, D., Oh, K.-J., and Choi, H.-J. (2017). The chatbot feels you-a counseling service using emotional response generation. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, pages 437–440. IEEE.

Lokhande, P., Aslam, F., Hawa, N., Munir, J., and Gulamgaus, M. (2015). Efficient way of web development using python and flask.

Mulvenna, M. D., Bond, R. R., Grigorash, A., O'Neill, S., and Ryan, A. Hilda-a health interaction log data analysis workflow to aid understanding of usage patterns and behaviours.

Ni, L., Lu, C., Liu, N., and Liu, J. (2017). Mandy: Towards a smart primary care chatbot application. In *International Symposium on Knowledge and Systems Sciences*, pages 38–52. Springer.

Tsiao, J. C.-s., Chao, D. Y., and Tong, P. P. (2007). Natural-language voice-activated personal assistant. US Patent 7,216,080.

Turing, A. (1950). Mind. *Mind*, 59(236):433–460.

Vīra, I., Teseļskis, J., and Skadiņa, I. (2014). Towards the development of the multilingual multimodal virtual agent. In *International Conference on Natural Language Processing*, pages 470–477. Springer.

Vogel, P., Klooster, T., Andrikopoulos, V., and Lungu, M. (2017). A low-effort analytics platform for visualizing evolving flask-based python web services. In *2017 IEEE Working Conference on Software Visualization (VISSOFT)*, pages 109–113. IEEE.

Xiaojiang, D. (2014). Chatbot system and method with contextual input and output messages. US Patent App. 13/661,040.

Yu, Z., Xu, Z., Black, A. W., and Rudnicky, A. (2016). Chatbot evaluation and database expansion via crowdsourcing. In *Proceedings of the chatbot workshop of LREC*, volume 63, page 102.