ALEXA, WHAT SHOULD I EAT? A PERSONALIZED VIRTUAL NUTRITION COACH FOR NATIVE AMERICAN DIABETES PATIENTS USING AMAZON'S SMART SPEAKER TECHNOLOGY

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Bikesh Maharjan

In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Software Engineering

November 2020

Fargo, North Dakota

North Dakota State University

Graduate School

Title

ALEXA, WHAT SHOULD I EAT? A PERSONALIZED VIRTUAL NUTRITION COACH FOR NATIVE AMERICAN DIABETES PATIENTS USING AMAZON'S SMART SPEAKER TECHNOLOGY

By

Bikesh Maharjan

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Juan Li

Chair

Dr. Jun Kong

Dr. Pinjing Zhao

Approved:

11/12/2020 — Dr. Simone Ludwig

Date — Department Chair

ABSTRACT

Among all other ethnic groups in the USA, native American's have higher chances of developing diabetes. A lot of tools have been developed to address this issue and help them in managing diabetes. However, these tools fail to address two major issues, first, the focus on research and need of Native Americans, and second, the intuitive user interface to use the functions available in these tools without requiring a complex knowledge of technology. This project focuses on reducing these two gaps. The project uses the underlying knowledge base to provide personalized recommendations and leverages the benefit of smart speakers to deliver the service to the user's which is highly intuitive and less demanding technologically. This project utilizes Ontology as a knowledge base and Amazon's Alexa platform for the initial experiment to provide personalized recommendations to the users.

ACKNOWLEDGEMENTS

I want to take this opportunity to thank my adviser Dr. Jen Li who has gone above and beyond to support throughout my graduate career. I am thankful for all the meetings and discussions we had during the research planning and progress discussions. All the things she has done to make me feel comfortable in the lab environment is a testimony for not only how great a professor and researcher she is, but also how good of a person and teammate she is to have in a project. Without her input and ideas during the discussion this project won't ever have completed, for that, I will always be indebted to her.

Also, I want to show my gratitude to Dr. Jun Kong who had found time in his busy schedule to be in the project meetings and provide his insights from his perspective. His ideas have helped in shaping this project over the period.

A special thanks to Dr. Pinjing Zhao, Professor in the Department of Chemistry at North Dakota State University for his consideration and finding time out of his busy schedule to be a part of my supervisory committee.

I have to applause my lab mates Zhuoran He, Vikram Pandey, and Shadi Alian for continuous support and brainstorming ideas as we worked on different projects.

My sincerest appreciation to all the staff and faculty of the Computer Science department for their role in continuous support and guidance throughout my career.

Finally, I express my gratitude to my friends and family for their continuous support as I navigate through my graduate career. The way they all stood strong together helped me focus on my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NDSU..North Dakota State University.

API ....Application Programming Interface.

AWS..Amazon Web Services.

AVS...Alexa Voice Service.

HIPAA ....Health Insurance Portability and Accountability Act.

JSON...JavaScript Object Notation.

IoT...Internet of Things.

BAA ....Business Associate Addendum/Agreement.

SDK...Software Development Kit.

AI ....Artificial Intelligence.

EHR...Electronic Health Record.

AVS...Amazon Voice Services.

ASK...Alexa Skills Kit.

ASR...Automatic Speech Recognition.

NLU ....Natural Language Understanding.

SDLC ....Software Development Life Cycle.

IDE..Integrated Development Environment.

JS...Java Script.

VUI ....Voice User Interface.

LTS ....Long Term Support.

VS ....Visual Studio.

HTTP...HyperText Transfer Protocol.

USDA..United States Department of Agriculture.

ADA .... American Diabetes Association.

IHS .... Indian Health Service.

SWRL .... Semantic Web Rule Language.

W3C .... World Wide Web Consortium.

OWL .... Web Ontology Language.

SUV... Sport Utility Vehicle.

REST... Representational State Transfer.

npm .... Node Package Manager

.

CHAPTER 1: INTRODUCTION

Diabetes has become one of the deadliest diseases in the world. Diabetes is one of the top 10 reasons for death globally [1]. Not only among diseases, but it is in the top 10 list of all the causes of deaths including road injury which demonstrates the lethal nature of diabetes. Back in 2000, diabetes was nowhere to be seen in this top 10 list [1]. In 2016 alone, 1.6 million people died from diabetes [1]. Developed countries out of all have more diabetes-related issues than its counterpart and are the fourth leading cause of deaths in developed nations [2]. One of the major effects of diabetes is it leads to lethal health conditions such as stroke, heart attack, blindness, kidney failure, amputations, and so on [3].

In the USA, 88 million adults have prediabetes and more than 8 out of 10 adults don't even know they have this condition [3]. Without proper medication, prediabetes can lead to type 2 diabetes in less than a few years. Today diabetes is the seventh leading cause of death in the USA and these data are still believed to be under-reported [3]. If the current trend continues 1 in 5 American's will have diabetes by 2025 [4]. Diabetes also has a huge economic impact for an individual as well as a nation. In the USA, the economic costs have increased by 26% from 2012 to 2017 in diabetes medication. In 2017, a total of $327 billion was spent on diabetes medication [5].

The condition is even worse for the Native American/Alaskan Natives (NA's) in the USA. Native Americans have higher chances of getting diabetes than any other ethnic group in the USA [1]. National Diabetes Statistic report of 2020 shows NAs have higher diabetic conditions in the classification list of diabetes in people 20 years and older, by race and ethnicity. 15.9 % of its total population has diabetes. 14.5 % of NA men and 14.8% of MA women have diabetes which is highest compared to any other race/ethnicity in the USA [6]. The incidence of diabetes was highest among the youths aged 10 – 19 years in the year 2014-2015 among NA's [7]. So, diabetes is no longer only the disease seen among the older aged NAs. In another research among 5274 Pima Indian children aged 5 to 19, each age group 5-9, 10-14, and 15-19 showed a significant increase in diabetic conditions by 1-2% [8].

Diabetes was very rare among Native Americans until the mid-twentieth century [9]. But later, diabetes has become a common, yet serious disease among the Native American tribes. A lot of research has been done to understand why diabetes has become such a common occurrence among Native American tribes, but it is not entirely clear. However, there has been constant research on a few of the determinants of diabetes. The first one is heredity. A full-blooded Native American showed the highest rate of diabetes than Half-Indian and other Non-Native Americans [10]. This study [11], which compared the blood sample of American Indians from Mandan, Arickara, and Hidatsa tribes with a White American and found the blood quantum of American Indians aged more than 35 showed a high prevalence of diabetes compared to the White Americans aged more than 35. Another reason is obesity. This study [12], concluded that low activity levels, high food intake, and socioeconomic status were major factors influencing obesity leading to diabetes in Native American's. The shift from the agricultural and hunting to an industrial economy has been referenced as one of the key reasons for diet change among Native American's [10]. Physical activity has decreased as people started using motorized vehicles for transport.

Studies have been done for possible options one can have to prevent, delay, and manage diabetes. Lifestyle modification with 7% in weight loss and 150 minutes of physical activity per week by non-diabetic people can reduce the chances of getting diabetes by 46% [13]. The daily decisions and habits of people with chronic diseases play a major role in their health condition.

Self-management is an important way to guide people's behavior and lead people to reach the desired goal and make the right decision [14]. A significant number of people when provided with their health status usually become more conscious of what they should eat and do to maintain good health. A study was done with Type 2 diabetic patients, where they were given their health status update and checked what they will choose as their behavioral goals. And It turns out a significant number of people chose good behavioral goals including an increase in physical activity, reduction in fat consumption, and increased fruit and vegetable consumption [15]. One research studied two brothers who have the same chronic problems inherent from their parents. However, one of the brothers regularly exercised, was concerned about his diet, and monitored blood pressure and glucose regularly which prevented him from getting the chronic disease [14]. Whereas, the other one did not exercise regularly and did not care about his diet which led to a gain in weight, and a hard time controlling his glucose level and blood pressure [14].

Appropriate food and nutrition are an important step for diabetes management [16]. The major obstacle here is being able to remember and apply all the complex health guidelines, constraints, and calculate calories manually. Currently, there are quite a few web and mobile applications that are available for calculating calories and act as a meal guide for people with diabetes [17], [18]. Such as with Go Meal [17] and Calorie King [18], one can keep track of what they ate using their smartphone and computer. However, these available applications are a) not well accepted among NA patients and b) do not use natural ways of communication such as voice conversations. Other few reasons these applications are not used among NAs are first these tools are not targeted specifically for Native Americans who have less exposure to health literacy, second is the cultural differences of NAs from others, and finally the complexity that exists in these digital tools, especially for the elderly population who have difficulties in using smartphone or computer because of poor vision or lack of experience in using such interface.

Voice assistants are the tools that can interpret human commands and respond using a natural language [19]. Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant are some of the widely used voice assistants which are also available as dedicated home speakers or embedded in smartphones and computing devices. The user base of this voice assistant is continuously rising. In January 2020 there were 87.7 million users of voice assistants which spiked from 66.4 million active users a year earlier[20]. Amazon is the leading in the market share of voice assistant use. 64.6% of voice assistant users use Amazon's Alexa followed by Google's assistant with 19.6% and Apple's Siri by 4.5% [20]. In one of the survey's done, most of the voice assistant users used their assistant for the music, search, and controlling the Internet of things (IoT) devices in their home [21].

To overcome the existing issues, we propose an Artificial Intelligence-driven voice agent to assist in diabetes self-management which is focused on the NA community. With a voice assistant, the user will be able to use natural language to give commands or have a conversation to get the calorie count and other recommendations. Voice is the natural form of communication that anyone can use with any level of technical expertise. But some challenges come with using voice assistants in the medical sector. The major challenges that are faced in introducing the voice assistants in the medical sector is having the technology that complies with the Health Insurance Portability and Accountability Act (HIPAA), and security and privacy issues on sharing personal information through a device [20].

With recent developments in the voice assistant's field, companies have started avenues to introduce HIPAA compliant applications through the voice platform and parallelly addressing privacy and security issues. Amazon's Alexa is one of the first voice assistants to provide HIPAA compliant platform and guidelines to develop voice applications that will comply with HIPAA [22]. So user can use Alexa skills which receive and transmit personal health information using the regulations defined in AWS Business Associate Addendum/ Agreement (BAA) [22]. Amazon whitepaper includes the following information which explains how HIPAA compliant platform is provided [23]. 1) The interaction between user and Alexa skill is not available to view through the web or mobile platform which provides security over patient health information 2) All the information that is used by HIPAA complaint skill are carried over by AWS servers which are HIPAA compliant and encrypts the user's information that conforms to guidelines from Secretary of Health and Human Services and 3) users have to retrieve their personal information rather than health professionals because of the rules which add an extra layer of security over users information. Currently, other available voice assistants such as Apple's HomePod and Google's assistant are not HIPAA compliant.

Amazon released its Alexa SDK which allows the developers to create voice skills according to their needs and reach out to its user base where they are, through a voice interface. Since then, many voice skills have been developed and uploaded to the skills store. Some skills that assist people with setting up alarms, making a to-do list, stream audiobooks or podcasts, or provide updates regarding traffic, weather, news, or other real-time events are already available through the skill store [24]. There are many HIPAA compliant skills which are made available in the skills store as well [25]. Some of the health skills are, Express Script is a skill that sends a notification to the people when their prescription has been shipped and when they are delivered. Atrium Health is another skill which allows user to find the urgent care locations near them and schedule appointments [25].

Researchers are continuously working to leverage the benefit of the voice interface in the healthcare domain. For an instance, some skills help elderly people to be able to do most of the usual tasks that we take for granted, such as being able to listen to the music of choice or news any time, being able to make their shopping list from home and even order items from the list [26]. Some skills help in performing the diagnosis of diseases such as colorectal cancers by asking a series of questions and then provide healthcare-related information to immunizations [27], [28].

Final few words on introducing the project that we worked on, this project focuses on using a voice assistant to help people with diabetes in the NA community in a culturally appropriate way. The project will use Amazon’s Alexa which will interact with the NA user and provide health education and meal recommendations. Users will have 24/7 access to the information. The recommendation of the meal will be based on their personal information such as age, disease, social setting, preferences, tastes, financial ability, literacy, and culture. And finally, it will remove any barrier that can exist because of the health literacy of NA users.

CHAPTER 2: RELATED WORK

Yum-me is the meal recommender that is used to study the personalized meal recommendation system [29]. Yum-me learns the user's preferences by using the visual quiz. Based on the response users provide, Yum-me selects the healthy meal that is most closely related to their preference. For an instance, in their research, they studied 3 people with no dietary restrictions and let them select the top-20 food items they prefer. Yum-me, according to their preference, selected two other healthy meals that match the user's preference. If we look at all the meal recommendation systems, most of the meal recommender system either used user ratings [30], [31], [32], [33], past recipe choices [34] and browsing history [34], [35] to come up with the recommendation. For an instance, [34] Designing and evaluating Kala's study used the past choices made by the users to recommend the recipe. Similarly, some research [30] used the user's ratings towards selecting the set of recipes and then calculate the nutritional requirement for each user.

In a study done by Ahmed Fadhil [36], he explores the limitations and complexities that exist in using chatbot or voice interface for meal recommendation. He explains the challenges that exist in the theoretical foundations of the voice assistant. He also explored the rule-based and AI bots to see any challenges there, explained the context limitation, logical flow, and language constraints that voice assistant can have. Another study experimented with the accuracy of the expert system developed for nutrition and calorie counting based on the voice description of food items against nutrition and calorie counting done by the dietician and found significant differences in the calorie counting done by the voice system compared to the one done by the dietician which clearly shows the need for improvement in the expert system [37].

Meal recommendation system

A wide range of applications is out there for the web and phone platform that helps to determine which meal to go for. Most of these applications focus on losing weight and providing good sources of nutrients to the users. There are applications such as Mealime [38] which provide recipes for meals to cook at home. These meals can be personalized for each user based on their preferences once they have created the account. Similarly, Platejoy [39] is another application using which user can have their personalized meal plans for a healthy lifestyle.

A meal plan is more important for a person who has diabetes. The food person eats is the driver that influences the blood sugar level of a person. So, a meal plan is a guide to determine what and how much to eat to maintain blood sugar within the target zone [40]. With this need, a lot of applications have been developed to focus on diabetic people. Some applications help keep track of diet and exercise. For example, MyFitnessPal [41] is an application that a user can use to keep track of calorie count and minutes of exercise he did. It also has a food database including restaurant foods. Another feature provided by this application is the ability to scan the bar code of the food items for fast and easy food tracking. There are other applications as well which provide similar functions such as MyDietDairy Calorie Counter [42], My Glucose Buddy [43], and My Dario Glucose Management System [44].

However, all these applications are focused on the phone or web platforms and don’t leverage the benefits of voice assistant technology. Moreover, these applications do not focus on the NA’s, and the differences that might exist such as literacy level, cultural and economic differences.

Voice in healthcare

Voice assistant has a wide range of usability in healthcare. Both patients and health care providers can take benefit from this. A voice assistant can be an answer to the patient who needs frequent health advice from the health care provider. A voice assistant can reduce the hours spent visiting the doctor by providing services 24/7. And for the health care providers, the voice assistant can be used for taking notes and documentation. And we are already seeing developments in these areas. There has been continuous research in this area to understand its usability in healthcare followed by the implementation. A software company called Naunce Communication has launched a voice assistant called Dragon Medical Virtual Assistant [45] which is an extension of their existing voice to text software. This assistant can document the communication and commands into Electronic Health Record (EHR) which saves time and money.

One of the studies researched the use of the voice assistant in helping people with surgery to answer their questions. This voice assistant is called Design, Development, and Evaluation of a Patient-Centered Health Dialog System to Support Inguinal Hernia Surgery Patient Information-Seeking [46] This dissertation research was done to build the use case for using a voice assistant to answer the questions that are raised after the hernia surgery. The assistant was built to take data collected by the patient as an input parameter and then reach out to a specific point for an answer. This research was focused to help people with low health literacy by using voice assistants such as Google Assistant and Amazon Alexa.

Alexa for healthcare

Amazon's Alexa is the first among the popular voice assistant's available in the market to launch the HIPAA compliant platform and skills to its users. Swedish Health Connect is a skill available using which users can make an appointment in the hospitals near them which are providing this service [47]. Similarly, Boston Children's Hospital has 'KidsMd' Alexa skill which allows parents to understand the symptoms that children are showing [48]. Another Skill, MyChildren's Enhanced Recovery, developed by Boston Children's Hospital lays a platform for parents to update the status of their children who have recently been through heart surgery. Another feature in the skill is the reminder of scheduled appointments [49]. Atrium Health is another skill that helps find the urgent care center nearby and help make an appointment. Users can also get information about the care center such as phone numbers and opening hours [50].

There exist some skills that focus on bringing people and health care closer. Such as skill which provides health care literacy to the people by explaining the medical terminology to the people. 'Answers By Cigna' [51] is a skill that focuses on this specific problem domain. Express Script [52] and Swedish Health Connect [47] are helping people to order their medicines from the comfort of their home and even make an appointment with their health care provider. Similarly, "Libertana Home Health Care" [53] reminds people of their medication and keeps track of the time when a user has taken the medication. This skill especially focuses the elderly users. The skill also creates a report later which is used by nurses to keep track of medication time. There are a lot of skills being developed every day in the health sector using Amazon Alexa. But there are quite a few skills that focus on helping diabetic people and even fewer application focusing NAs with diabetes.

CHAPTER 3: BACKGROUND TECHNOLOGY ABOUT ALEXA

Amazon released Alexa first time back in 2014 to the users embedded in Amazon's Echo [54]. Back then it was the first of its kind voice assistant available in the market to the general public. Alexa works by listening to the voice commands from the user and then figuring out if it can follow the command to get the work done. Alexa has a wake word which will trigger it to start listening to the user's command. I will talk more about how the complete request is processed in detail in the coming chapters. But the high-level overview will be, the entire process starts with the user giving a command to Alexa, which will send the command to cloud services. A cloud service will then process and generate the response and return the response to the user. In the next few paragraphs, I will talk about a few technologies that are either used by Alexa or influences the Voice technology platform.

Alexa Skills Kit (ASK) is the Software Development Kit (SDK) that the team in Amazon released to the general public so that who wants to develop an Alexa skill can use this to build their own Alexa skills which can be made available to the wide general public which people then can access in the Alexa enabled devices. Developers can use the SDK found in different development languages such as Java, C#, Python, and Node.js. The SDK helps the developer to quickly use various features it provides to create voice solutions and release it in the Amazon Skills pool. ASK SDK provides already built-in skill templates that help the developer get the skill work quickly done without having to write a lot of code. Another major benefit of SDK is it has built-in support for handling the request coming from the device and sending a response back to the device, so anyone developing the skill does not have to worry about how to deserialize the request and how to package the response so that it is properly received in the other end. All that is already pre-preprogrammed within Alexa and developer can only focus on the functionality of the skill. Besides that, SDK also provides a platform to save the attributes for a short time or the lifetime of the skill. This helps to remember what has happened before any moment during the conversation with the user.

Alexa Voice Service is another service that is released right after Alexa which allows the hardware builders to integrate Alexa into their device and call the services on the cloud to integrate with the Alexa service. This service is very important in Amazon's plan because it will allow Alexa to be used in a wide range of devices which can in turn make it (Alexa) a de-facto service that comes with any hardware one might purchase. With AVS users can interact with hardware using the voice. Amazon provides a wide range of AVS ranging from chipset helping device makers to include hardware that is capable of voice processing, or hardware that is capable of calling the APIs to provide the service.

When I talk about these technologies behind Alexa, I must talk about Amazon Lex. Amazon Lex is the service Amazon provided after Alexa is released using which one can develop conversational bot such as Google's assistant, Apple's Siri as well as Amazon's Alexa itself. The benefit of using Amazon Lex is it helps in Automatic Speech Recognition (ASR) and understand what the conversation is trying to say or understand its intent using the Natural Language Understanding (NLU) features. This service helps the developer to create their chatbots to reach out to the user domain without following the rules set for Amazon Alexa.

Above, when I wrote about the templates provided by the ASK SDK, I did not go into details about each one. Let's talk about a few because one of them is used during the prototype development later in the implementation chapter. Smart Home Skill is one of the templates which generates the code for the Alexa skill to communicate with home devices such as coffee maker, oven, or thermostat. This template helps by generating the possible intents that come into play when developing a skill that controls the home device. Similarly, Game skills templates provide the code block required to create interactive trivia skills. But the one we are interested in is the custom skill template. A custom skill template is the one used for prototype development later in the implementation chapter. The reason to select the custom skill template is with a custom skill, we only get the basic functionality that the skill will need such as what to do when a command could not be fulfilled and gives a blank paper fill up to create a skill as we need.

CHAPTER 4: SYSTEM DESIGN

The software development life cycle (SDLC) chosen for this research is the classic waterfall model. The reasons for choosing this development methodology was clear. We had a defined set of requirements, which would not change until the completion of the project. And we had the defined process of how to complete the project. Because of fewer unknowns at the beginning of the project waterfall seemed the most appropriate to go with.

The platform chosen for this project as mentioned above sections is Amazon's Alexa. Alexa being the first platform to provide HIPAA compliance is one of the major reasons to go with Alexa. But besides that it is the first platform that released its Software Development Kit (SDK) to the developers, using which developers can not only develop the skill they want but also make it available to the general public easily with release options available through Amazon Web Services (AWS). The integration was seamless because developers would develop the skill conforming to the SDK.

The following figure [Figure 1] shows the overall process that occurs within Amazon Alexa's conversation. Alexa has a frontend and backend. The frontend is a Voice User Interface (VUI), though with a user can interact with Alexa, ask questions or give commands, and the backend consists of the code that handles each conversation scenario and forms a response speech to be spoken back to the user. A skill which is developed for Amazon Alexa usually consist of specific functions that it can handle, these functions or features are known as Intentions in the Alexa. And users can activate or ask questions based on each intention. The speeches are already defined in the Alexa which will trigger each intention. These speeches are known as utterances.

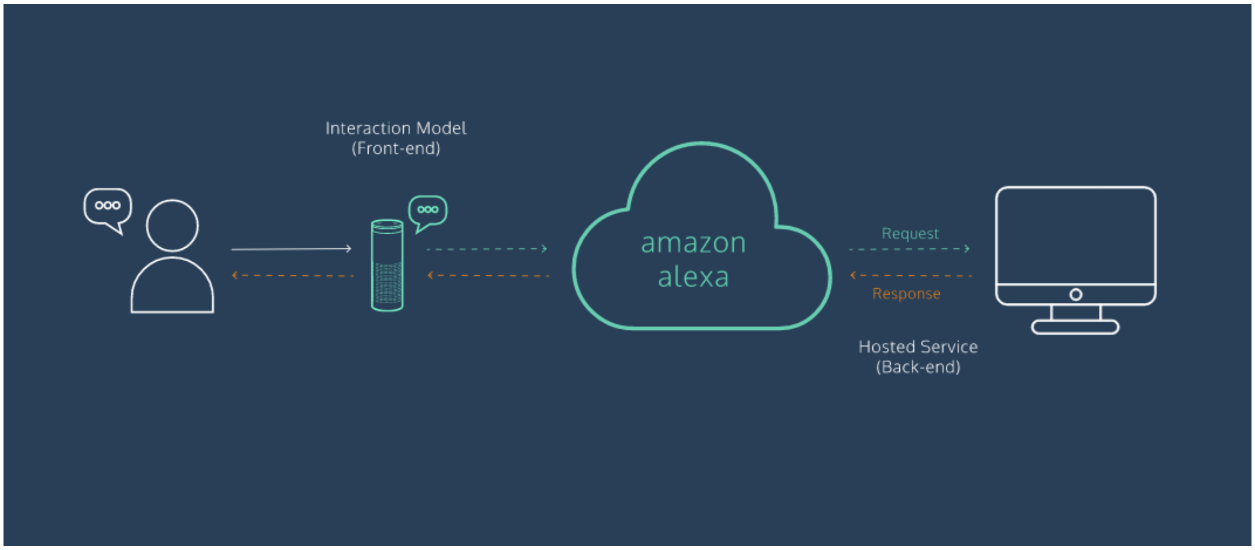

Figure 1. Dialog flow between user and Alexa[55]

Alexa Developer Console was used to design and develop the front-end VUI of the skill. In the Alexa Developer Console, VUI could be developed using two ways. The first way is to use simple button clicks to add each intent and utterances and other functionality to design the VUI and second, there is a JavaScript Object Notation (JSON) editor which can be used to design the conversation (intents and utterances). In this skill development, we first used the simple button clicks to set up the initial structure of the VUI and then used the JSON editor to continue editing our frontend as required for the skill. After all, the entire frontend of the Alexa is just a JSON which is transferred over the Alexa backend and AWS. For the backend development, we used the node JS version 8.10 Long Term Supported (LTS) version. Alexa SDK for Node JS [56] was used to develop the skill that will conform with the Alexa skill development.

All the development was done in the local environment for Alexa Backend. A Node Package Manager (npm) package, Alexa-skill-local [57] was used to host the backend and make it available to be used by the Alexa frontend. For all of this to work, we have to log in using the

Amazon developer’s credentials to be able to use this npm package. Most of the backend code was developed using the Visual Studio Code (VS Code) editor.

Another important aspect of this project is the Application Programming Interface (API) which connects the requests coming from the Alexa backend, deserializes the data, feeds the data into the reasoner, reasoner then processes the data and selects the optimal meal options and sends it back to the API, API will form a response and sends it to the Alexa backend which is finally spoken back to the user as a final response. This API is developed using the Java programming language. Javalin [58] is used as an API development framework. It is a lightweight web development framework for Java and Kotlin. IntelliJ Integrated Development Environment (IDE) is used to code and develop the backend API. And finally, we had the ontology that will contain all the information for NA person, diabetes guidelines, food, and nutrition information.

So, the plan was to develop each part and connect to get the complete functionality.

Overall design

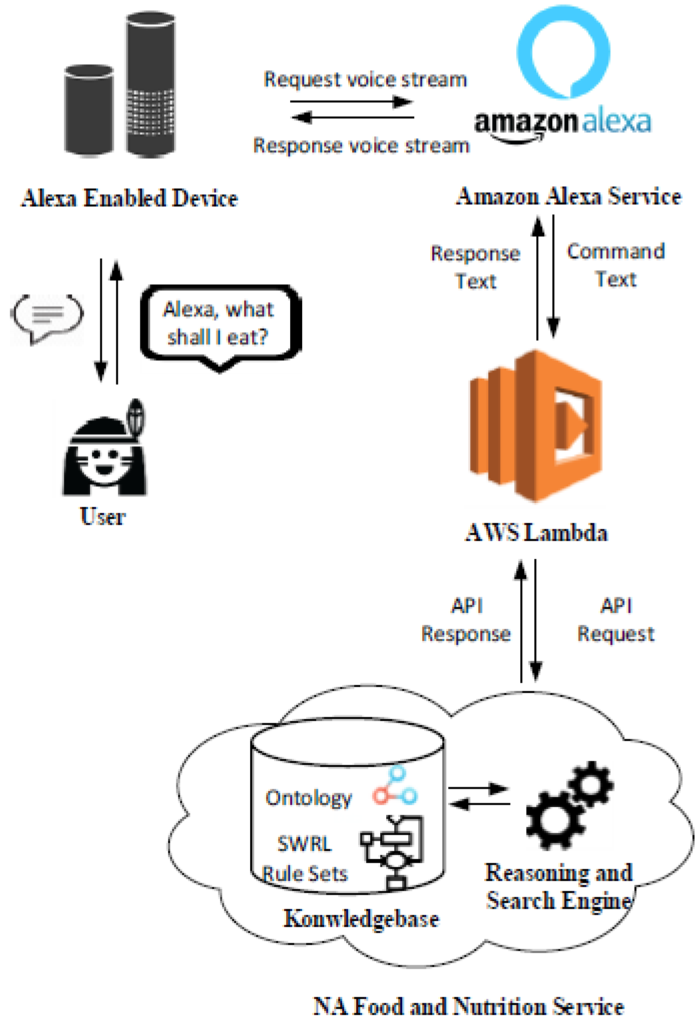

Figure 2. The overall architecture of the system [59]

The major goal of this system was to serve as an assistant for the NA user to help manage their diabetes by recommending the optimal meals and provide health information to increase their health literacy when and where needed. The above figure [Figure 2] shows the overall system architecture of how the conversation will occur and what parts come into play for a successful conversation between user and Alexa.

Our user will be a native American who will require help with a meal recommendation and will be using our skill to get the information required. The initial assumption for this user is, the user will be a diabetic patient and has intermediate to low health literacy. The user will interact with the skill through an Alexa enabled hardware device, which can be a dedicated Alexa enabled devices such as echo dot, Amazon Echo, etc., any smartphone which has Alexa enabled or a computer with Alexa. The user will be triggering the conversation with the skill with the dedicated speech phrase for the skill such as “What do you recommend for the dinner” or “what shall I eat?”.

Once Skill is activated, it connects it to the Skill backend which handles the conversation. The Alexa backend is developed using the Alexa SDK which provides the benefit to the developers by directly converting the speech that the user provides into the programmable JSON objects. So, developers don’t need to handle the underneath work of understanding what the user is speaking. It is handled by the Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) functionality that comes along with the SDK. ASR is a technology that Amazon uses to map the speech into text [60]. Amazon refers to this as a first step that enables voice-driven technology such as Alexa. ASR has the capability of understanding what the user is speaking and convert it into words. SDK comes with its various features [60] that can be used to develop more interactive skills for the users. Among all the features, Entity Resolution and Dialog Management are the features that are used the most in this skill. A more detailed description of how these are used within the skill are explained in the later part of the disucssion.

As mentioned above, Alexa has a front end and the back end. Let’s first talk about the frontend we have in our system. The Alexa front end comprises of various intents (features and functions of skill) that we want to provide through our skill. We implemented a few intents that will help a user to get the information they need for a meal or nutrition to manage diabetes. An example of an intent we have in our voice skill is explaining to the user about nutrition. For an instance, if the user asks " what is fiber?" then our skill will handle that through an intent "Nutrition Explanation Intent" or if a user asks "recommend me a meal" then that conversation will be handled by a different intent "Meal Recommendation Intent". In short, intents are like the function blocks that will serve one purpose to return the response to the user for a specific command. Within each intent, there are utterances in Alexa. Utterances are the possible phrases that a user might say to start the conversation or in response to a continued conversation. Utterances have slots that are defined within them. Slots are like variable placeholders to read the important data required to make a decision. Slots will read the data as the conversation is carried on between Alexa and NA users. The reading of the data from the conversation and storing it in the slots are all handled by the SDK with the help of its underlying ASR and NLU capabilities. During the entire conversation, Alexa back end decides which slots are already collected and which still needs to be collected. Alexa backend is developed using node JS and uses npm package Alexa-skill local to host it to connect with Alexa frontend which we discussed earlier. Based on the decision made, Alexa backed drives the conversation by returning the response that will motivate the user to respond in a way that will help to assign a slot with an acceptable value. Alexa backend processes the conversation that will lead to the success of the conversation with a collection of all the required data points.

Once all the data points are collected by the Alexa backend. It will make an HTTP (HyperText Transfer Protocol) call to the API that is developed using JAVA, more precisely Javalin Framework as mentioned above. The API reads the data sent, makes any necessary conversion to the data as required before making a call to a reasoner. It is more like a validation part where it pre-processes the received data. Then finally this received data is used to make an informed decision based on rules that are in the reasoner to select the set of meals that will meet the criteria of providing the required nutrients for the patient and which will not affect their health negatively.

Once these meals are selected, they will be sent back to the API. Once again API will perform any necessary processing to create the response required to be sent to the Alexa backend. This response is sent to the Alexa backend in the form of JSON. Using this JSON which has an array of food items, including the nutrition facts of each food item. Finally, the Alexa backend will form the response based on the number of food items received. The Alexa backend will cache these food items and its details and send one response at a time to the Alexa frontend until the conversation is over.

Not all the conversation needs to go to the ontology or backend database or persistent storage that is placed to store the data all the time. In our case, it is when the user asks for the health information, for an instance " what is fiber? ", in this case, if the API does not have this information with it cached, then only it will reach the backend persistent storage which has this information. Otherwise, it will just read it from the cache without having to call the backend storage every time.

Alexa's frontend, finally as an endpoint on the user side, speaks the responses sent to it from the Alexa backend and collects the required data from the user side during the conversation.

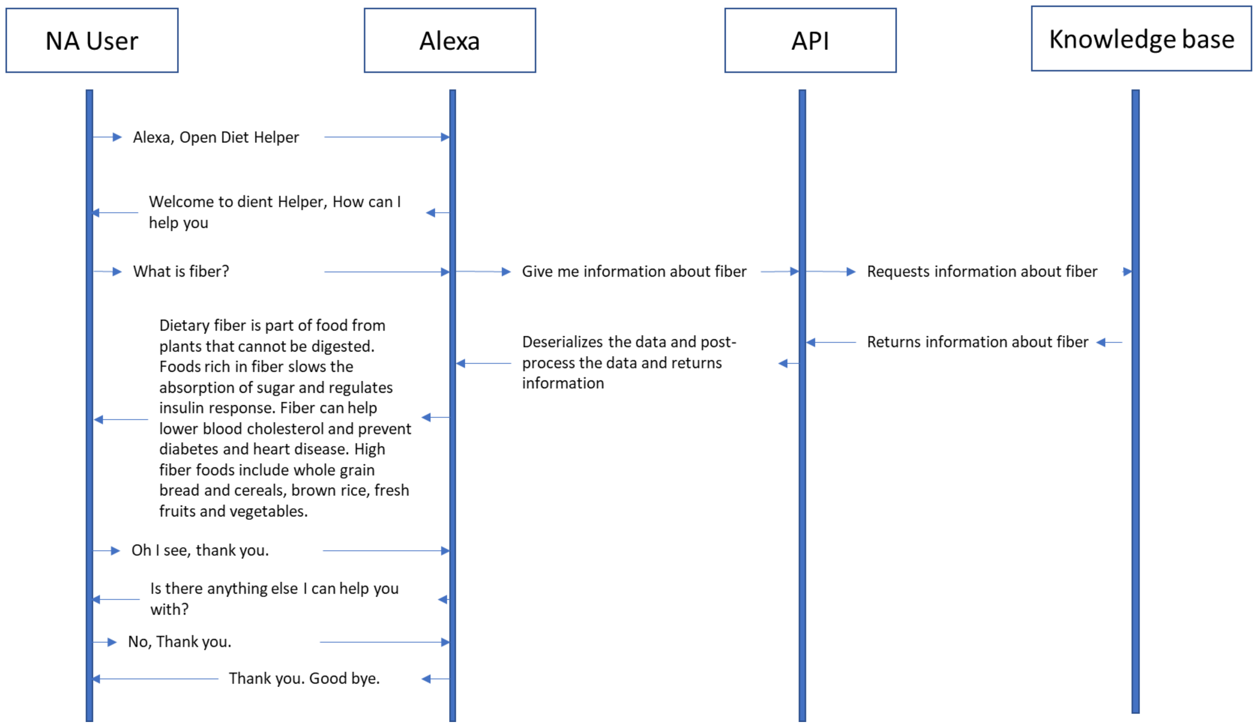

Figure 3. Sequence diagram showing the flow of conversation between NA user, Alexa, API, and Knowledge base

CHAPTER 5: PROTOTYPE IMPLEMENTATION

Part 1: Extending domain knowledge of Alexa

Most of the health care applications that exist in the voice interface have less to limited knowledge of the domain. In addition to that, when it comes to providing recommendations to the specific group/community of the people it's even less. At the beginning of the project when we searched for applications that focused on specific needs and requirements of a NA user was not found. So, our voice assistant's major base stands on having domain-specific knowledge in form of ontology and using that knowledge to provide personalized recommendations. To add more value to our recommendation we have rules that are based on dietary guidelines from the United States Department of Agriculture (USDA) and the American Diabetes Association (ADA). All the testing is done in the laboratory environment and no human users were involved during the testing of the prototype.

Knowledge in the form of ontology

Ontologies are used in the computational domain which deals with modeling the structure of the system that includes entities and relations that emerge between these entities [61]. For an instance, an ontology for a company with its employees will consist of a hierarchy of employees that exist and their responsibilities. Using an ontology, we can define entities, their properties, and the relationship between these entities that makes sense to a human. An ontology is human readable and does not require the need to have high-level knowledge of the system to understand the relationships as we see in the relational databases. It defines domain knowledge in a standard way that allows computer programs to understand it and query as well [62].

We used an ontology-based system to store food and nutrition information. Ontology also consisted of NA profile and guidelines for diabetes. Guidelines for diabetes were used to generate the semantic rules to generate recommendations. With all the information available in an ontology, it will extend Alexa's knowledge in the domain of diabetes and food recommendation rules for the NA user. The first part of the ontology consists of the NA user profile which captures the socio-economic properties of the user, personal health information using which we can provide personalized recommendations. And the second part of the ontology consists of diabetes information, diabetes guidelines, diabetes education, and food and nutrition information. The figure [Figure 4] shows the part of the ontology design used in this system.

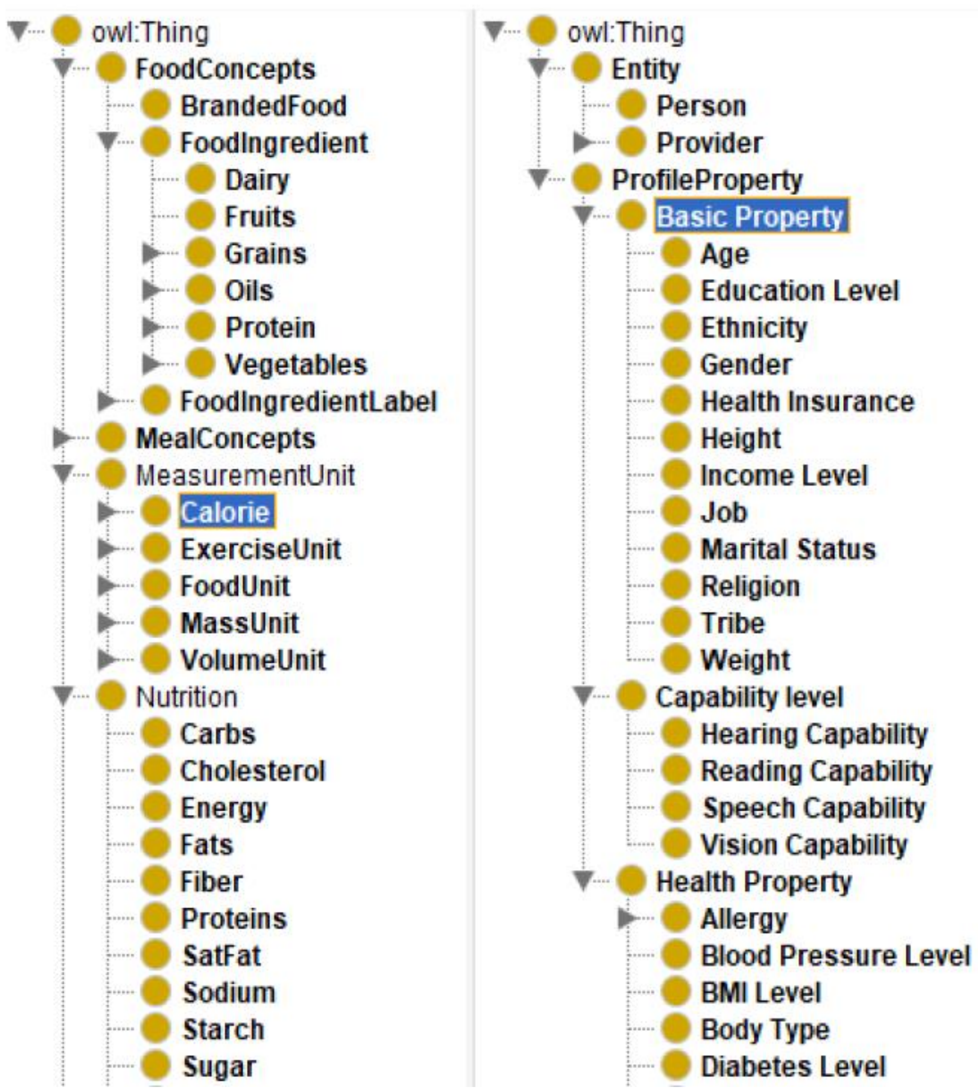

Figure 4. Part of classes used in ontology to define the knowledge base

Health rules used by reasoner

Like we said earlier the other part of the ontology consist of the semantic rules which are based on the medical guidelines for the management of diabetes. These medical guidelines for the management of diabetes are collected from the resources such as the American Diabetes Association (ADA) [63], Nutritional Recommendation for Individuals with Diabetes [64], the prevention and control of Type -2 diabetes by changing lifestyle [65], USDA's guidelines for the management of diabetes [66], IHS [67], American Indian and Alaska Health [67] and the National Indian Health Board [68]. These rules were first collected and then verified by the physicians of our research team. Finally, these guidelines were made rules that a computer can understand and will be personalized to the NA user. These rules are written using the Semantic Web Rule Language (SWRL) which is a World Wide Web Consortium (W3C) standard Web Ontology Language (OWL). After these rules are available in ontology, any program can query the ontology for the recommendation of the meals and based on these rules ontology is able to send back the set of meals which meet the criteria.

The profile maintained in the ontology has sub-profiles, a) basic profile which stores the user's basic information such as age, gender, and education background b) health profile which stores users health information such as diabetes, allergies, and medication c) Social profile will store user's religion and income and d) Preference profile. The user profile will store information such as daily blood sugar level, physical activity, and the meals user had each day.

Part 2: Handling dialog with a user

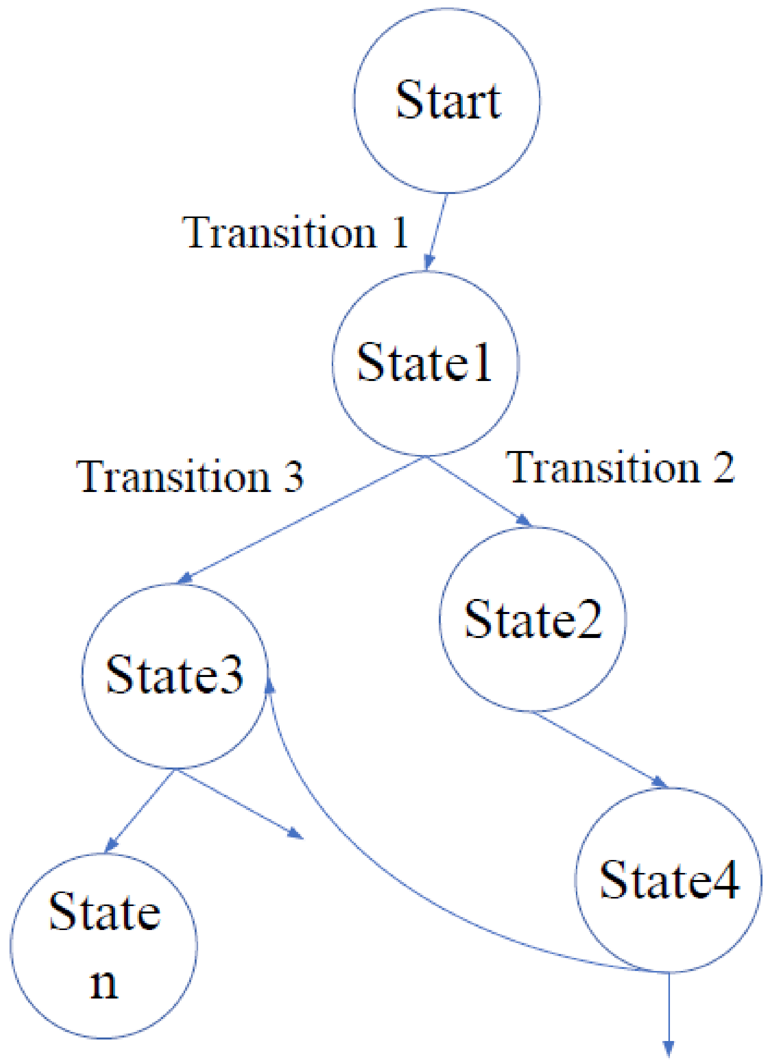

Figure 5. State transition to keep track of the dialog

For the dialog design and directing the conversation to the direction of completion, we selected a state-based dialog model. We followed this design to model a dialog between the user and Alexa. The advantage of the state-based dialog design is being able to track the flow of the conversation and being able to stay in the context of the conversation. For an instance, a conversation is going on between the user and Alexa to get information to buy a bicycle and in between the conversation, the user drifts from the current conversation to get information about the mountain bike, which lies outside of the bicycle buying context. When this happens, we can remember where the conversation paused for buying a bicycle and started getting information about mountain bikes. So, that we can jump back into the previous conversation once the user is done with getting the required information. So, state design helps in keeping the historical information of the conversation which serves as a memory for our dialog design. And this is what happens in the natural conversation between two people, we can remember the context of the conversation.

The state at which the current conversation is in will be based on the information that has been received so far from the user to complete the dialog successfully between the user and Alexa. Figure [Figure 5] above provides the graphical components of how a state will change within a conversation between Alexa and the user. But let’s take a conversation that will happen between Alexa and the user for our analogy of how the state will change during the conversation. Usually, the state is in an initial state/ start state when the user starts a conversation by asking a question. For our analogy, we will take an example where the user is trying to get a recommendation of a dog (Start State). Once the conversation has started; the next state will change when Alexa is trying to get specific information from the user before it can recommend a dog to a user. Suppose Alexa is trying to get the values for two qualities that the user is looking for in a dog (GetQuality State). Until the user has not provided the values for two qualities that the user is looking for in a dog, this state will not change. Once the user has provided the information for the qualities, they are looking for in a dog, the state will again change. Maybe this time, Alexa, needs to know the size of the dog the user is looking for. So, let’s say the conversation has reached (GetSize State). Do you see how state change takes place in the state-based model? And finally, after the user provides the size of the dog user is looking for Alexa will provide the recommendation of the dog which we can take as the final state of this conversation design (Final State).

So, our state graph will include a. a start state which represents the starting of a conversation, possible input messages from the user, based on these input data, we will have a possible set of states that will change and finally the possible conclusion based on the direction of the conversation or state changes. Another major benefit of the state-driven conversation is it helps keep the conversation within a similar context as much as possible which guarantees to conclude the conversation with a higher percentage of success.

We leveraged the benefits of the dialog management feature from the Alexa ASK [69] to help control the state of the conversation. Dialog management makes it easy to create a multi-turn conversation in Alexa. We can define what are the values that are required to be obtained from the user to continue the conversation in the Alexa front-end and then using dialog management we can easily drive the conversation to obtain these values before we move to the next state. For an instance, we are making a skill for a car shop, where we first need to get the values if the user wants a sedan or an SUV. If the user says sedan, then we need to provide a compact or full-size sedan. Similarly, if a user selects SUV then we might want to ask sports or family hauler. Dialog management helps in getting those values and driving the conversation based on the obtained value. Using the dialog managements, STARTED, IN PROGRESS, and COMPLETED state, we can control what Alexa will say next to drive the conversation and move closer to the result.

During this conversation, the information that is obtained can be used for further personalization of the results for the next time. The values currently missing in the ontology can be saved if is obtained during the conversation such as the user’s height, weight. Similarly, the food items that user-selected quite frequently can be used to create a favorite meal entity in our ontology and add properties to it. But most importantly, the data received during the conversation is helpful to remember the context of the conversation. Even if the user changes the context of the conversation,

Here I am going to discuss a little about what intents, utterances, and slots are.

Intents

Intents can be taken as a function that an Alexa skill can provide services to. In other words, this can be taken as a specific task that the user is trying to accomplish. In this implementation, the intents are intentions we want to serve through the skill, which is, meal recommendation, recipe recommendation, and nutrition education. Intent gives a high-level view of the functions that a skill supports right now. Some skills might be a music player in which possible intents could be PauseIntent, PlayIntent, NextIntent, PreviousIntent, and so forth.

Utterance

An utterance is the specific phrases that users will speak to start a request to Alexa. This is also a trigger, that is used to differentiate which intent should Alexa start. This is the major part of the dialog design that needs more planning and constant development because there are multiple ways people might ask the same thing. For an instance, a user is asking to get a fact from Alexa, the following are the various ways a user might ask for the information

"Give me a fact"
"Tell me a fact"
"Tell me a diabetes fact"
"Give me a diabetes fact"
"What's within a diabetes fact"

So, the utterance is the area major dialog design is focused around since a user might say the same thing differently every single time.

Slots

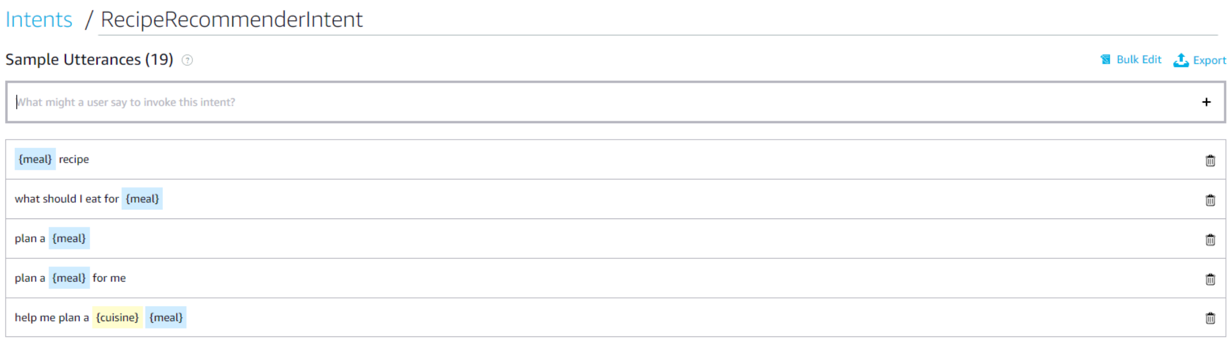

Figure 6. The intent, utterance, and slots

The slot is a variable that will store the value for the specific intent. The slots help Alexa figure out if the initial requirement from the user is complete or not to take the conversation ahead. The above figure [Figure 6] shows the basic intent, utterance, and slots at the same place in the Alexa frontend design. From the figure, we can see, the intent name is RecipeRecommenderIntent, which means it will focus on recommending a recipe to the users. Utterances are the phrases that are defined which will help Alexa figure out which intent to trigger. In this case, “Plan a {meal}” is one of the utterances that a user might say to start this intent. And finally, a slot is the variable {meal} that lives within an utterance to store possible values. In this scenario, it might be “breakfast”, “lunch” or “dinner”.

There are built-in slot types in the ASK SDK that one can use, but it also allows users to create the custom slots which make Alexa even more powerful from the usability standpoint. One can even define what will be a similar value for the corresponding slot value. These are known as synonyms in the Alexa front end. When similar words are spoken during the utterance, it will be mapped to the same value. For example, users might say “affordable”, “low priced”, “cheap” or “economical” for the low-priced item. And these all can be mapped to the same value in the Alexa front-end.

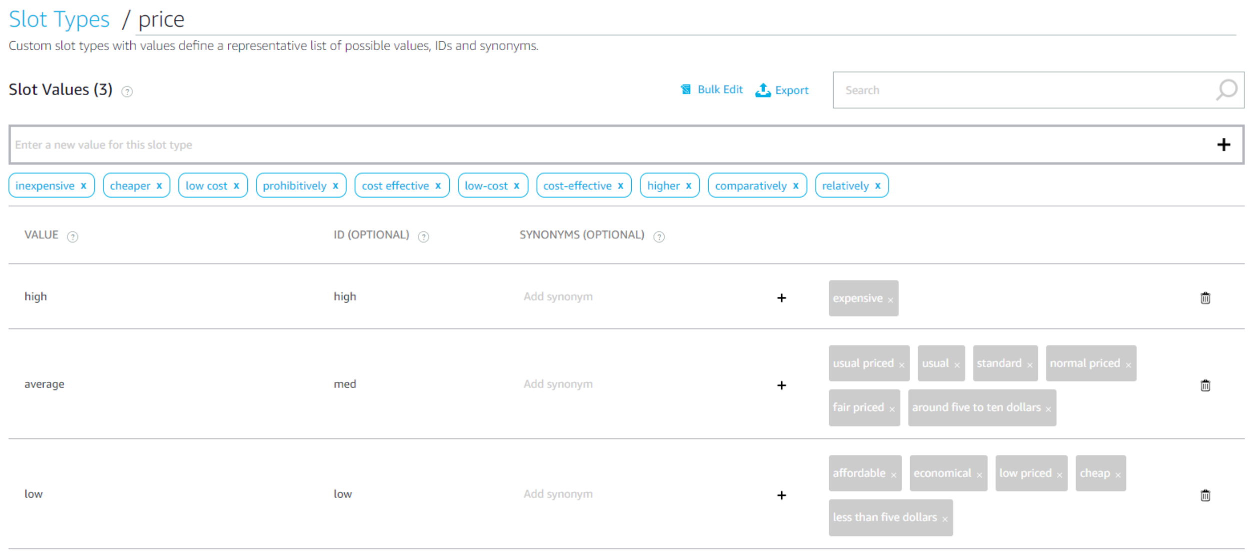

Figure 7. The synonym implementation in Alexa frontend

The whole Alexa front-end interaction model is a JSON file.

Part 3: Generating response

Although the dialog management of the Alexa ASK SDK can handle responses during the conversation, some response requires sending the request to the API which queries the ontology. A response is generated based on the context in which a user is asking a question. Based on the intent of the conversation, dialog management generates the next response. But to get the recommendation for the meals, a request is sent to the API and API queries the ontology that we talked about above. Once the API sends back the data, Alexa-backend forms the response to be spoken back to the user from the Alexa frontend.

To clear out the above paragraph let's go into how, where, and when’s of the response generation. Like said above, the first place where the response is generated is the Alexa itself which is handled by the dialog management, one of the great perks of using ASK SDK. For some of the general responses which don’t require a lot of calculations or need to call other API or services, Alexa can generate the response to send back to the user based on the command user is giving. For an instance, the welcome message to give to the user when skill is invoked the first time doesn't need to query the complex API / services. We can write possible welcome messages to give users when we design the Alexa.

User: Alexa, open Diet Helper.
Alexa: Welcome to Diet Helper….

So, that's one response that Alexa can generate based on what we have set in the Alexa for our skill. It is possible to set a wide range of messages and then tell Alexa to randomly select from the list of messages as well. Another response which Alexa, can handle for us is a conversation within the context to get the required information from the user. What I mean by this is, for a dialog to complete Alexa has to provide information that the user is seeking, but even before that can happen Alexa needs to get the complete information or get the required data. When we talked about slots above, Alexa needs to get the values of the required slots to send the request to the higher-level services (API) to get the response. The responses generated until these slot values are calculated can also be generated by Alexa front and back end. We can set these responses both in Alexa front and back end and program Alexa to select the response as the conversation goes on.

Alexa stores the information in the following JSON format.

Another important feature the ASK SDK provides when Alexa is collecting the data for each slot is the validation. We can validate if the value that the user provides falls into the set of values that satisfy the criteria for that specific slot. For an instance, we have a slot that reads the name of the planet. For this slot, we can set the validation rule that will accept the name that matches one of the planet names we have in our solar system. But if the user provides a different name besides the name of the planets, we can catch that using the validation rules we can set in the Alexa and handle the response accordingly. Consider the following scenario,

Alexa: Which plant do you want to know about?
User: Tell me about the Sun.
Alexa: I apologize but Sun doesn't fall in the category of planets. Can you tell me the name of a planet?


```
"prompts": [
    {
        "id": "Elicit.Slot.1160489731214.702724138893",
        "variations": [
            {
                "type": "PlainText",
                "value": "Can you please tell me which one would you like, veg or non veg?"
            },
            {
                "type": "PlainText",
                "value": "Would you prefer vegeterian or non vegeterian?"
            },
            {
                "type": "PlainText",
                "value": "There is vegeterian and non vegeterian breakfast, which one would you prefer?"
            }
        ]
    },
```


Figure 8. Alexa storing response variation in the form of JSON

So, these were all the responses that are handled in the Alexa front and backend. But the response that could not be handled in the Alexa front or back end needs a call to the API that uses the data and reasoner to generate the response for the query. The details of the API and how it is developed is coming up in the Application Programming Interface part next. In this section, we will discuss how the API comes into play when generating a response.

Once the required slot values are saved by the Alexa and it needs to get more complex calculations and reasoning done, then it will call the API by sending the HTTP request. HTTP request will contain the required Header and Body. API is always running in the remote server, ready to receive any request and process it. Once the HTTP request is received, API calls the reasoner which uses the knowledge base to get an answer, and based on that reply a response data is prepared in the API and sent back through the HTTP response message. And finally, that message is sent back to the user.

Part 4: Application Programming Interface

API acts as a middleware between the ontology and the Alexa. API is a simple REST API that is developed using JAVA. API uses the Javalin [58] web framework. This API is used by the mobile application that is developed as part of the implementation of the research as well. Which I am not going to talk about in detail.

API plays a very important role in the functioning of the skill development and completion of the conversation with the user. API helps to keep the health rules and reasoning logic independent of the Alexa skill which helps in separation of concerns in the side that interact with the user and the side that interacts with the data. After the completion, there is not much to show on the application side but most of the time during the implantation is spent in the development of the API.

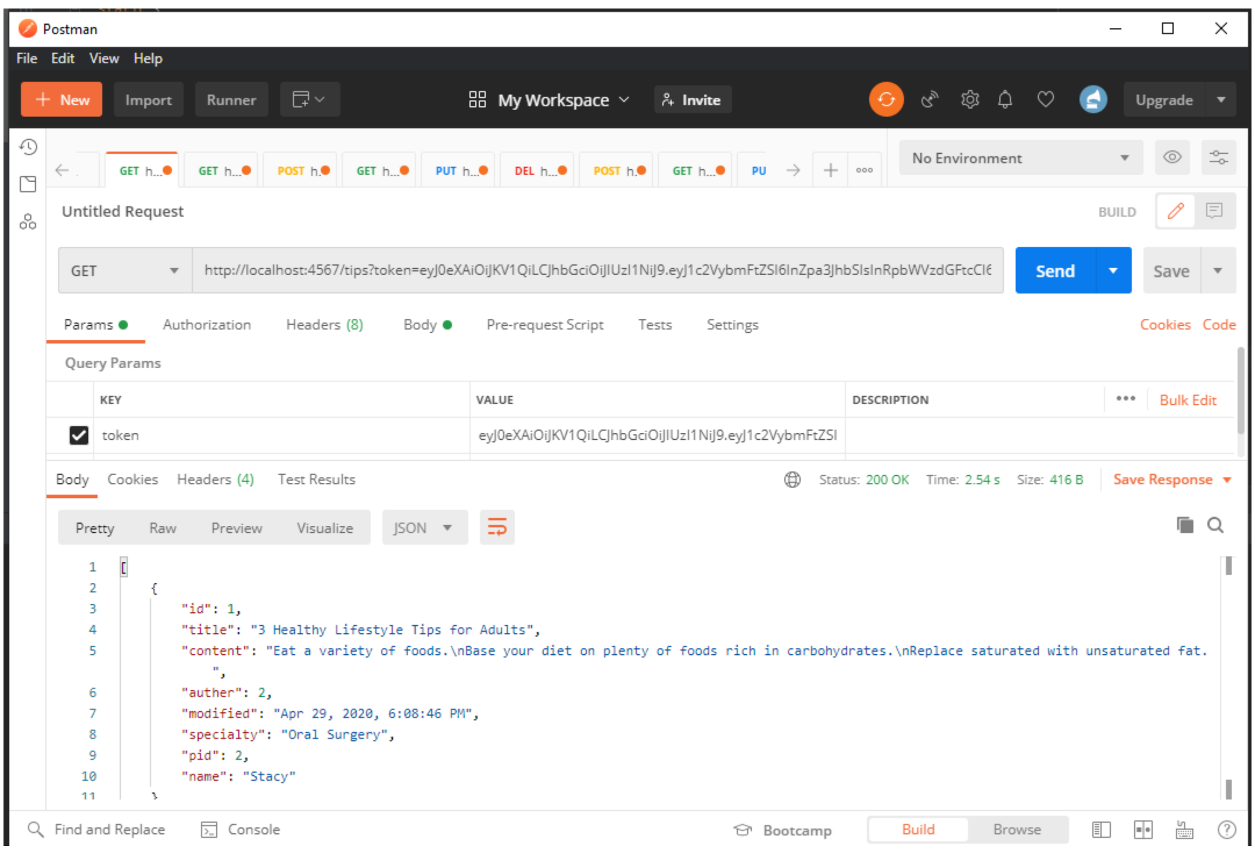

Figure 9. Postman tool clip showing how data is returned when requested in the JSON format

Figure [Figure 9] shows the functioning of the API that is deployed in the server. Alexa backend will make a similar call to the API to get the required data. Here is a response [Figure 10] that is received in the API for the list of vegetables with all the data available, calculations are made for the meal recommendation when the user needs help in meal recommendation.

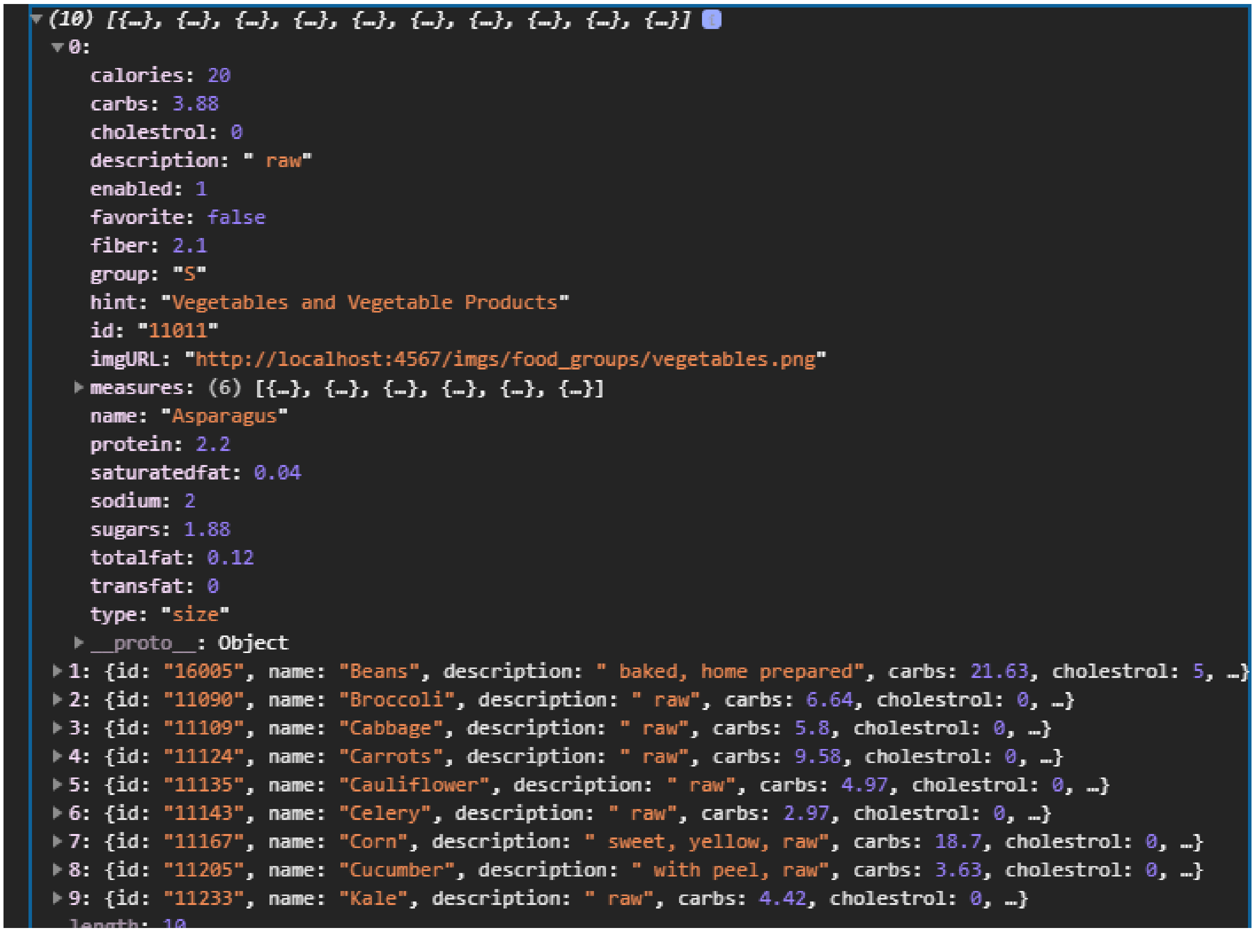

Figure 10. Details of food items being returned from a Knowledge base

CHAPTER 6: EVALUATION AND DISCUSSION

Use cases

The skill that's developed can handle multiple use cases, intents in the language of the skill. The first one is recommending meals to the user based on their profile, the second one is recipe recommendation which sends the recipe to the user phone and also speaks out the required ingredients, the third one is providing nutrition education to the user when the user asks about nutrition, the fourth one is a local restaurant and restaurant menu recommendation, the fifth one is diet history recall in daily and weekly period, the sixth one is saving the food items into the favorite list of user and the seventh one is daily/weekly diet summary and comments. Based on how the conversation flows and what is the intent of the conversation, multiple functionalities can work together to provide information to the user.

Giving nutrition education

This intent is activated when the user wants to learn about nutrition and what is the meaning of nutrition such as fiber. The following figure [Figure 11] shows how this intent help understands the user about nutrition. In the whole conversation, Alexa helps the user with the calculation of calories in a diet and provides brief nutrition information.

The skill is activated when the user asks Alexa to start a skill. Once the skill is activated, Alexa will greet the user with an opening statement letting the user know that skill is ready to handle the queries from the user. The user asks to help in counting the calories in food items he is thinking of. Using the knowledge base, and the help of API, Alexa counts the calorie and responds with total calorie information. Following that user wants to know if the food is good for him or not. Here in this part, another intent is activated which is the recommendation of the meal. Again, with the help of ontology, which consists of the user's personal information, and back end

API, if the food is good for the user is decided. And the response is created to be spoken back to the user.

User: Hi Alexa, Open Diet Helper.
Alexa: Welcome to Diet Helper. What can I do for you?
User: How many calories are in 2 eggs and 4 slices of bacon?
Alexa: 2 large boiled eggs have about 156 calories and 4 slices of bacon have about 178 calories, so all together it's 334 calories.
User: Is it good for breakfast?
Alexa: It is fine. But I would suggest adding some food rich in fiber.
User: What do you mean by fiber?
Alexa: Dietary fiber is the part of food from plants that cannot be digested. Foods rich in fiber slows the absorption of sugar and regulates insulin response. Fiber can help lower blood cholesterol and prevent diabetes and heart disease. High fiber food include: whole grain breads and cereals, brown rice, fresh fruits and vegetables.
User: Thank you.
Alexa: Is there anything else I can help?
User: No, thanks.
Alexa: Okay. Goodbye.

Figure 11. Dialog between the user and Alexa to get the nutrition information

Then in the following conversation, the user asks about the fiber which activates the intent that activates the nutrition information. This nutrition information is available in the backend ontology and also in the cache in the API since the nutrition information is less likely to change. Once nutrition information is retrieved, it is explained back to the user through the Alexa frontend. And finally, the end of the conversation occurs when the user doesn't need any help from Alexa.

The above conversation scenario demonstrates not only intent that explains the nutrition to the user, but also how other intents activate as the conversation goes on. Similarly, we can see call the API and ontology happen multiple times to retrieve the user's information and food nutrition information. The use of the caching in the API side improves the response time which stores the information such as nutrition details and others which are less likely to change quickly.

Recipe recommendation

User: Hi Alexa, Open Diet Helper.
Alexa: Welcome to Diet Helper. What can I do for you?
User: Help me with a meal.
Alexa: Sure, What do you crave?
User: Something spicy.
Alexa: I see, you want some spicy food. How much cooking time do you have?
User: Probably an hour.
Alexa: Do you want some vegies or meat item?
User: I was planning for some meat.
Alexa: Great. How about spicy Mongolian Beef sounds?
User: No, I don't think so.
Alexa: No problem. How about Spicy Mexican Chicken Salad?
User: sounds good. But how to make it?
Alexa: I will tell you the ingredients and check the link I sent you to you phone as well. You can ask me to repeat anytime. Are you ready?
User: Yes.

Figure 12. Dialog between the user and Alexa for a recipe recommendation

The second intent that the skill can handle is recommending the recipe to the users. The above figure [Figure 12] shows an overview of the conversation that a user has with Alexa to get the recommendation of the recipe. Users can activate this intent by saying multiple utterances (speech) that will activate this meal recommendation intent. Once the intent is activated, it will try to get a few information from the user before being able to recommend any meal. Some of the slot data like in the above scenario are spiciness level, vegetarian or non-vegetarian meal, duration of time that a user has to prepare a meal. Once all these parameters are received, Alexa will call the API which will query the knowledge base and then get the list of the meals. Once the list is processed, few items at a time are sent to the Alexa front-end. Once Alexa receives the list of meals, it will speak the name of the meal to see if the user is interested.

In case the user likes the meal, it will provide the user with ingredients that are required to prepare the meal and send a final request to the API that dialog has completed so that API can drop the remaining list of items.

In case the user doesn't like the meal, Alexa will provide the name of another meal until the user selects one meal. Once all the meals are exhausted, Alexa will send a request to the API to retrieve another list of possible meals. This process will continue until the user selects one meal or decides to stop getting the recommendation. Either way, Alexa will send an acknowledgment to API which helps clear up the list that API might be holding up.

Meal recommendation

As discussed in scenario 1, which has the main function of providing education to the user about the nutrition, meal recommendation was embedded as well. In this intent, it sums up the nutrients and deciding if the meal is good or bad. For an instance, when the user asks Is 2 cups of rice and 1 cup of beans is good for me? Then Alexa sends data to the API in the following format:

```json
{
  "items": [
    {
      "food": "rice",
      "unit": "cups",
      "value": "2"
    },
    {
      "food": "beans",
      "unit": "cups",
      "value": 1
    }
  ]
}
```

API deserializes that information and then calculates the nutrition values using the knowledge base. Each person has their recommended limit saved in their health profile. The following figure [Figure 13] shows, how API is getting back the personal recommendation saved in the health profile. And that profile is used to make a recommendation.

```
▼{userId: 8, userGender: 1, userPhysicalActivityLevel: 1, userBloodPressureLevel: 2, userDiabetesType: 1, …}
 ▶allergies: []
  bmi: 24.365970597095576
  bmiLevel: 1
  eer: 2403.634014976
  email: "sandra@gmail.com"
  isUserAlcoholic: 0
  isUserHasBloodPressure: true
  isUserHasDiabetes: true
  isUserSmoker: 0
  maxCPM: 4
  maxCarbs: 240
  maxCholeterol: 205
  maxEnergy: 2403.634014976
  maxFat: 93.47465613795555
  maxProtein: 180.27255112319997
  maxSatFat: 26.707044610844445
  maxSodium: 2300
  maxSugar: 60.0908503744
  maxTransFat: 2.6707044610844446
  minCPM: 3
  minCarbs: 180
  minCholesterol: 180
  minFat: 53.41408922168889
  minFiber: 33.650876209664
  minProtein: 120.1817007488
  name: "sandra"
  pac: 1.12
  userAddress: undefined
  userAge: 26
  userBloodPressureLevel: 2
  userBodyType: undefined
 ▶userDOB: Sat Feb 26 1994 00:00:00 GMT-0500 (Eastern Standard Time) {}
  userDiabetesLevel: 1
  userDiabetesType: 1
  userEducationLevel: 1
  userGender: 1
  userHeight: 69
  userHip: 0
  userId: 8
  userPhysicalActivityLevel: 1
  userTribe: 0
  userWaist: 0
  userWeight: 165
  username: "sandra"
 ▶__proto__: Object
```


Figure 13. Knowledge base returning the personal information of a user with their health information

Analysis

Here we are going to discuss the data obtained after the experiment was performed for the following purposes. Purpose 1 is recipe recommendation, purpose 2 is food and nutrition recommendation, and purpose 3 is a daily nutrition summary.

The data is calculated based on the total of 150 conversations attempted for each of the intent. For these experiments, the conversation was carried out only for one intent. So, these data do not show the data for the conversations performed for nested intents.

Definition of success

If the user gets the complete response by the time the conversation is completed, it is considered a successful conversation. If the user doesn't get the desired response during the conversation because of the error in the code or skill not being able to understand what the user is trying to query, that is considered a failure. One important factor to consider is, for this experiment, a maximum threshold was set for the number of turns that can occur during a conversation, to force to reach the desired conclusion as soon as possible. If the skill could not respond before this threshold is met, then it is considered a failure. Also, we checked in each successful conversation, the response provided by the skill was accurate or not.

So, the success rate is calculated using the following formula,

$$\text{success rate} = \frac{\#\text{ of successful conversation}}{\#\text{ of total conversations}}$$

The following table [Table 1] lists the experiment results. The accuracy is calculated using the following formula.

$$\text{accuracy} = \frac{\#\text{ of correct recommendations}}{\#\text{ of total recommendations}}$$

Results

Table 1. Results of the experiment

Metrics	Purpose 1	Purpose 2	Purpose 3
Dialog Success rate	76%	87.0%	80.0%
Avg Dialog time	69.4s	43.8s	45.3s
Time per turn	6.1s	8.4s	8.57s
Number of turns per dialog	11.0	5.2	5.6
Recommendation accuracy (success conversation)	100%	100%	100%

From the table, we can see that the success rate ranges from 76 to 87 percentage. Most of the failure rate occurred when Alexa could not identify what was the user's intent. An average conversation time was around 43 to 70 seconds based on what was the user's intent. The trend that was common after the experiment was, longer the conversation took place, they were more likely to fail than the short period of conversation. Longer conversations are usually result when Alexa took more time to understand the user intent and the data that the user-provided.

From the skill side, we tried to include as many utterances as we can and add more directions to help the user to encourage the user to provide the required information. Among the successful recommendations, the accuracy was 100 percent. All the recommendations followed the required medical guidelines and rules.

CHAPTER 7: CONCLUSION AND FUTURE WORKS

Conclusion

This research presents the preliminary results of the possible use of the voice interface and technology in the field of healthcare. Using voice is the most natural means to communicate which doesn't require any prior experience and exposure in using a certain technology. This research focuses on using the voice skill developed in the Amazon Alexa platform to help the NA user. He learned the usefulness and power of a combination of the knowledgebase, ASK SDK, and the middleware API to provide personalized recommendations to the users.

Alexa being HIPAA compliant provides all the developers and health care providers another means to reach people in need and provide the service through the means of voice. We experimented with few use cases that could apply to a NA user to help them manage their diabetes. But there sure can be more use cases than this which will vary according to the scope at which skill is targeted to be used.

We targeted the NA user because of the high percentage of the diabetic patient among the NA and the help that's available is not focused on the NA user. With more personalized recommendations we believe this mechanism can make more effective in helping the user in managing their diabetes.

Future works

The future involves planning on how we can improve the success rate of the conversation and learn more use cases as we plan on deploying the system, we have developed to the NA community and learn the effectiveness of the system.

There are few ideas already floating around on how we can improve the success rate of the conversation. The first one is for the error's where the Alexa is not able to match the user's queries to any of the available intents, the improvement on how we will help the user to go in the direction to ask a question in a way in which an Alexa can map the query to an intent which is already there. And the second one is, for every query that Alexa was not able to map it to a specific query, send it to the backend API to be saved in a database somewhere. So, all the queries that failed could be analyzed further and incorporated in the Alexa front/back end, so that Alexa will be able to map almost every query to specific intent and we will be able to provide more services.

The saved queries that failed will also allow us to analyze and find use-cases that users of the skills are looking for and later include it as a part of a skill or create an entirely different skill.

REFERENCES

[1] WHO, “Top 10 causes of death.” [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (accessed Apr. 10, 2020).

[2] R. Torkington-Stokes, D. Metcalf, and P. Bowler, “Management of diabetic foot ulcers: Evaluation of case studies,” *Br. J. Nurs.*, vol. 25, no. 15, pp. S27–S33, 2016, doi: 10.12968/bjon.2016.25.15.S27.

[3] CDC, “About Prediabetes & Type 2 Diabetes.” [Online]. Available: https://www.cdc.gov/diabetes/prevention/about-prediabetes.html.

[4] J. P. Boyle, T. J. Thompson, E. W. Gregg, L. E. Barker, and D. F. Williamson, “Projection of the year 2050 burden of diabetes in the US adult population: dynamic modeling of incidence, mortality, and prediabetes prevalence,” *Popul. Health Metr.*, vol. 8, no. 1, p. 29, 2010, doi: 10.1186/1478-7954-8-29.

[5] “Economic Costs of Diabetes in the U.S. in 2017,” *Diabetes Care*, vol. 41, no. 5, pp. 917–928, 2018, doi: 10.2337/dci18-0007.

[6] CDC, “National Diabetes Statistics Report, 2020,” *Natl. Diabetes Stat. Rep.*, p. 2, 2020, [Online]. Available: https://www.cdc.gov/diabetes/pdfs/data/statistics/national-diabetes-statistics-report.pdf.

[7] D. Dabelea *et al.*, “Association of Type 1 Diabetes vs Type 2 Diabetes Diagnosed During Childhood and Adolescence With Complications During Teenage Years and Young Adulthood.,” *JAMA*, vol. 317, no. 8, pp. 825–835, Feb. 2017, doi: 10.1001/jama.2017.0686.

[8] D. Dabelea, R. L. Hanson, P. H. Bennett, J. Roumain, W. C. Knowler, and D. J. Pettitt, “Increasing prevalence of Type II diabetes in American Indian children.,” *Diabetologia*, vol. 41, no. 8, pp. 904–910, Aug. 1998, doi: 10.1007/s001250051006.

[9] M. B. Hoy, “Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants,” *Med. Ref. Serv. Q.*, vol. 37, no. 1, pp. 81–88, Jan. 2018, doi: 10.1080/02763869.2018.1404391.

[10] D. Gohdes, “Diabetes in North American Indians and Alaska Natives,” *Diabetes Am. 2nd ed.*, vol. 211, no. 7, pp. 683–701, 1995, [Online]. Available: http://diabetes.niddk.nih.gov/dm/pubs/america.

[11] J. D. Brosseau, “Diabetes among the three affiliated tribes: Correlation with degree of Indian inheritance,” *Am. J. Public Health*, vol. 69, no. 1979, pp. 1277–1278, 1979.

[12] K. M. West, “Culture, history, and adiposity, or should Santa Claus reduce? Obesity and Bariatric Medicine,” vol. 3, no. 1974, pp. 48–52, 1974.

[13] W. C. Knowler *et al.*, “Reduction in the incidence of type 2 diabetes with lifestyle intervention or metformin,” *N. Engl. J. Med.*, vol. 346, no. 6, pp. 393–403, 2002, doi: 10.1056/NEJMoa012512.

[14] T. Bodenheimer and K. Lorig, “BodenheimerT_Patient Self-management of Chronic Disease in Primary Care — JAMA,” vol. 288, no. 19, pp. 2469–2475, 2002.

[15] P. A. Estabrooks *et al.*, “The Frequency and Behavioral Outcomes of Goal Choices in the Self-management of Diabetes,” *Diabetes Educ.*, vol. 31, no. 3, pp. 391–400, May 2005, doi: 10.1177/0145721705276578.

[16] “Nutrition Recommendations and Principles for People With Diabetes Mellitus,” *Diabetes Care*, vol. 17, no. 5, pp. 519–522, 1994, doi: 10.2337/diacare.17.5.519.

[17] P. Hudson “Go Meals.” [Online]. Available: http://www.diabetesincontrol.com/gomeals/ (accessed Sep. 10, 2020).

[18] K. McGuinness, “Calorie King.” [Online]. Available: https://www.calorieking.com/ (accessed Sep. 10, 2020).

[19] M. B. Hoy, “Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants,” *Med. Ref. Serv. Q.*, vol. 37, no. 1, pp. 81–88, Jan. 2018, doi: 10.1080/02763869.2018.1404391.

[20] Voicebot.ai, “the Voice Assistant Consumer Adoption Report 2018,” no. November, pp. 1–33, 2018, [Online]. Available: https://voicebot.ai/wp-content/uploads/2019/01/voice-assistant-consumer-adoption-report-2018-voicebot.pdf.

[21] T. Ammari, J. Kaye, J. Y. Tsai, and F. Bentley, “Music, Search, and IoT: How People (Really) Use Voice Assistants,” *ACM Trans. Comput. Interact.*, vol. 26, no. 3, Apr. 2019, doi: 10.1145/3311956.

[22] Amazon, “Introducing New Alexa Healthcare Skills.” https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2019/04/introducing-new-alexa-healthcare-skills (accessed Apr. 10, 2020).

[23] AWS, “Architecting for HIPAA Security and Compliance on Amazon Web Services Amazon Web Services – Architecting for HIPAA Security and Compliance on Amazon Web Services,” no. August, 2015, [Online]. Available: https://aws.amazon.com/compliance/hipaa-compliance/ (accessed Sep. 10, 2020).

[24] Wikipedia, “Amazon Alexa.” [Online]. Available: https://en.wikipedia.org/wiki/Amazon_Alexa (accessed Sep. 10, 2020).

[25] S. Alder, “Amazon Announces 6 New HIPAA Compliant Alexa Skills.” [Online]. Available: https://www.hipaajournal.com/hipaa-compliant-alexa-skills/ (accessed Sep. 10, 2020).

[26] F. Capan, “Why Amazon device is a gift for healthcare What WWE taught me about EHR promotion,” Viewpoint, 2016, [Online]. Avalilable: https://media.mmm-online.com/documents/188/january_2016_viewpoint_46793.pdf

[27] E. Savitz, “How Virtual Health Assistants Can Reshape Healthcare.” 2013, [Online]. Available: https://www.forbes.com/sites/ciocentral/2013/03/13/how-virtual-health-assistants-can-reshape-healthcare/?sh=5e926e7f2000

[28] M. Parfitt, “How AI driven tech is being used to assist older people.” [Online]. Available: https://www.system-concepts.com/insights/voice-interfaces-in-home-healthcare/ (accessed Apr. 10, 2020).

[29] L. Yang *et al.*, “Yum-Me: A personalized nutrient-based meal recommender system,” *ACM Trans. Inf. Syst.*, vol. 36, no. 1, 2017, doi: 10.1145/3072614.

[30] D. Elsweiler and M. Harvey, “Towards Automatic Meal Plan Recommendations for Balanced Nutrition,” in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 313–316, doi: 10.1145/2792838.2799665.

[31] H. Y. Chang, D. X. Li, Q. D. Liu, R. J. Hu, and R. S. Zhang, “Content-enhanced matrix factorization for recommender systems,” *Appl. Mech. Mater.*, vol. 475–476, pp. 1084–1089, 2014, doi: 10.4028/www.scientific.net/AMM.475-476.1084.

[32] J. Freyne and S. Berkovsky, “Intelligent Food Planning: Personalized Recipe Recommendation,” in *Proceedings of the 15th International Conference on Intelligent User Interfaces*, 2010, pp. 321–324, doi: 10.1145/1719970.1720021.

[33] M. Harvey, B. Ludwig, and D. Elsweiler, "You Are What You Eat: Learning User Tastes for Rating Prediction," in *String Processing and Information Retrieval*, 2013, pp. 153–164.

[34] M. Svensson, K. Höök, and R. Cöster, *Designing and evaluating kalas: A social navigation system for food recipes.*, vol. 12. 2005.

[35] Y. van Pinxteren, G. Geleijnse, and P. Kamsteeg, "Deriving a Recipe Similarity Measure for Recommending Healthful Meals," in *Proceedings of the 16th International Conference on Intelligent User Interfaces*, 2011, pp. 105–114, doi: 10.1145/1943403.1943422.

[36] A. Fadhil, "Can a Chatbot Determine My Diet?: Addressing Challenges of Chatbot Application for Meal Recommendation," 2018, [Online]. Available: http://arxiv.org/abs/1802.09100.

[37] P. Ladyzynski, J. Krzymien, P. Foltynski, M. Rachuta, and B. Bonalska, "Accuracy of automatic carbohydrate, protein, fat and calorie counting based on voice descriptions of meals in people with type 1 diabetes," *Nutrients*, vol. 10, no. 4, 2018, doi: 10.3390/nu10040518.

[38] J. Bunn, "Meal lime." [Online]. Available: https://www.mealime.com/.

[39] C. Bognet, "Plate Joy." [Online]. Available: https://www.platejoy.com/.

[40] CDC, "Diabetes Meal Planning." [Online]. Available: https://www.cdc.gov/diabetes/managing/eat-well/meal-plan-method.html.

[41] P. Frisk, "My fitness plan." [Online]. Available: https://www.myfitnesspal.com/.

[42] C. Thompson, "My diet dairy." [Online]. Available: https://www.medhelp.org/land/calorie-counter-app.

[43] T. Xu, “My Glucose buddy.” [Online]. Available: https://www.glucosebuddy.com/.

[44] E. Raphael, “My dario glucose management system.” [Online]. Available: https://www.mydario.com.

[45] Nuance, “New Virtual Assistant Capabilities Enhance Interactions between Clinicians and Patients, Improving Patient Experience and Reducing Physician Burnout.”, 2018, [Online]. Available: https://www.nuance.com/en-gb/about-us/newsroom/press-releases/nuance-unveils-AI-Powered-solution-for-healthcare.html.

[46] R. J. Lordon and S. Munson, “Design, Development, and Evaluation of a Patient-Centered Health Dialog System to Support Inguinal Hernia Surgery Patient Information-Seeking,” 2019, [Online]. Available: https://digital.lib.washington.edu/researchworks/handle/1773/44675.

[47] Amazon, “Swedish Health Connect.” [Online]. Available: https://www.amazon.com/DIT-Providence-St-Joseph-Health/dp/B07PGJYYF6.

[48] J. Comstock, “Kids Md.” [Online]. Available: https://www.healthcareitnews.com/news/boston-childrens-hospital-launches-amazon-alexa-app-kidsmd.

[49] Amazon, “My Children’s Enchanced Recovery After Surgery.” [Online]. Available: https://www.amazon.com/MyChildrens-Enhanced-Recovery-After-Surgery/dp/B07QB7PQYW.

[50] Amazon, “Atrium Health.” [Online]. Available: https://www.amazon.com/Carolinas-Healthcare-Atrium-Health/dp/B07H1T46DC.

[51] Cigna, “Answers by cigna.” [Online]. Available: https://www.cigna.com/newsroom/news-releases/2018/cigna-enters-voice-control-space-launching-answers-by-cigna-skill-for-

amazon-alexa.

[52] E. Script, “Express script.” [Online]. Available: https://www.express-scripts.com/corporate/articles/amazons-alexa-helps-members-check-their-prescription-order-status-and-more.

[53] Libertana, “Libertana Home Health Care.” [Online]. Available: https://www.libertana.com/.

[54] D. Etherington, “Amazon Echo Is a $199 Connected Speaker Packing An Always-On Siri-Style Assistant,” *TechCrunch*. https://techcrunch.com/2014/11/06/amazon-echo/?guccounter=1&guce_referrer=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw&guce_referrer_sig=AQAAAEWNFdPmHJYNs2fYSZxZ0VB-z-IqAJXPMX9o7Rmr1nlALbhM0CKKxGXaO1AbiLfSgCVEHpT3ly2BE-NpOLeNaIbAZS_YvHBmlpWUrj_AoYpjXr8Cy8JB4T2EsqxY.

[55] J. Jeffress, “Advanced Skill Building with Dialog Management,” 2018, [Online]. Available: https://m.media-amazon.com/images/G/01/mobile-apps/dex/alexa/alexa-skills-kit/guide/Dialog-Management-Guide.pdf?aliId=26037891.

[56] Amazon, “ASK SDK for Node Js.” [Online]. Available: https://developer.amazon.com/en-US/docs/alexa/alexa-skills-kit-sdk-for-nodejs/overview.html.

[57] A. Milmile, “Alexa Skill Local.” [Online]. Available: https://www.npmjs.com/package/alexa-skill-local.

[58] Javalin, “javalin.” [Online]. Available: https://www.javalin.io.

[59] B. Maharjan, J. Li, J. Kong, and C. Tao, “Alexa, What Should I Eat? : A Personalized Virtual Nutrition Coach for Native American Diabetes Patients Using Amazon’s Smart Speaker Technology,” *2019 IEEE Int. Conf. E-Health Networking, Appl. Serv. Heal.*

2019, 2019, doi: 10.1109/HealthCom46333.2019.9009613.

[60] Amazon, “What is automatic speech recognition?” [Online]. Available: https://developer.amazon.com/en-US/alexa/alexa-skills-kit/asr.

[61] N. Guarino, D. Oberle, and S. Staab, “What Is an Ontology?,” in *Handbook on Ontologies*, 2009, pp. 1–17.

[62] J. Li and S. Alian, “Design and Development of a Biocultural Ontology for Personalized Diabetes Self-Management of American Indians,” in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2018, pp. 1–7, doi: 10.1109/HealthCom.2018.8531107.

[63] M. Care, “Standards of Medical Care in Diabetes—2018,” vol. 41, no. January, pp. 1–24, 2018.

[64] N. Silvis, “Nutritional recommendations for individuals with diabetes mellitus,” *South African Med. J.*, vol. 81, no. 3, pp. 162–166, 1992.

[65] M. Asif, “The prevention and control the type-2 diabetes by changing lifestyle and dietary pattern,” *J. Educ. Health Promot.*, vol. 3, p. 1, Feb. 2014, doi: 10.4103/2277-9531.127541.

[66] U.S. Department of Health and Human Services, “2015-2020 Dietary Guidelines.” [Online]. Available: https://health.gov/our-work/food-nutrition/2015-2020-dietary-guidelines.

[67] U.S. Department of Health and Human Services, “Indian Health Service.” [Online]. Available: https://www.ihs.gov/.

[68] National Indian Health Board, “National Indian Health Board and Public Health.” [Online]. Available: https://www.nihb.org/public_health/public_health.php.

[69] A. Kumar, A. Gupta, J. Chan, S. Tucker, B. Hoffmeister, and M. Dreyer, “Just ASK: Building an Architecture for Extensible Self-Service Spoken Language Understanding,” Nov. 2017.