Project Report

Voice-Based Database Search Tool

Master of Science
in
Computer Science

Submitted by

Jasminder Singh Issar

Under the guidance of
Dr. Vijay Mago

Department of Computer Science
Lakehead University
Thunder Bay, Ontario, Canada

January 16, 2019

Abstract

In this new era of technological advancement of human kind, we have made a vast number of advancements and exchange enormous amount of data every passing minute of a day. Nearly ”2.5 quintillion bytes of data have been created worldwide in each day of 2017” [1]. It can be a very difficult task to convert the data into useful information. To address this issue, a voice service with the help of AWS framework is used to make a tool which can make the data more useful and prediction easier. The voice service used is Alexa in the application and we are using Lambda service to write our back-end code. Our application totally works using cloud-based services. A voice-based application which can converse with user is made and it acts as a database search tool which can filter the data on the basis of user input. With the advancement in Artificial Intelligence and Machine learning, there is a significant growth in Natural language processing and that makes our machines really smart and convenient to interact with users [2]. NLP is the cognitive computing component focused on linguistics and language's classification and makes Alexa more easy to understand human spoken words. It helps the machines to learn fast with time. This application is mostly data independent and will work on any CSV file of some particular format. There is also a one page interface created to store the file in S3 database, so that Alexa will be able to use it and all this happens in real time. Further user can save those filtered data in data file in S3 and can view it later on.

Acknowledgments

I would like to express my sincere gratitude to my final year project supervisor, Dr. Vijay Mago, whose contribution in stimulating suggestions and encouragement, helped me in all the time of research and writing of this project report.

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this report.

I am thankful to my parents for telling me what I'm capable of and for giving me the support that I needed to build a dream to chase after. I'm grateful to Lakehead University for providing me such an environment to reach my goals and to prosper in the corporate world.

Jasminder Singh Issar

January 16, 2019
Lakehead University

Contents

List of Figures

Chapter 1

Introduction

The reason why AI or Artificial Intelligence has been affordable to everyone is due to a different varieties of open source software which has allowed the building of advanced self-learning systems. ”TensorFlow built by Google is an example of such open source software. Presently, TensorFlow is being used by a huge number of companies who provide AI consulting services. Some of them include Dropbox, eBay, Intel, Twitter and Uber” [3]. It is considered as or classified among the most precise machine learning frameworks. Keras, which ”is another library for building deep learning models, is built on top of TensorFlow and allows to quickly build and test a neural network with minimal lines of code” [3].

”Natural language processing (NLP) is the ability of a computer program to understand” [4] and interpret human language as it is spoken and is a component of artificial intelligence. NLP applications are slow in development. It is challenging because computers traditionally require humans to ”speak” to them in a programming language that is precise, unambiguous and highly structured, or through a limited number of clearly enunciated voice commands. Human speech, however, is not always precise – it is often ambiguous, and the linguistic structure can depend on many complex variables, including slang, regional dialects and social context [5]. The machines which use AI and are powered by machine learning algorithms need to be trained using large labeled datasets. But supplying training data for specific situations is not always possible, and the process itself can be quite expensive. We generally do not need all data when using reinforcement learning. Reinforcement learning opens great opportunities for building AI applications with general-use deep learning algorithms [3]. We are seeing more intelligent and natural conversations between voice assistants like Alexa, Google Home. With the advancements in AI, such systems learn to speak more instinctively [6].

In this era, we are surrounded by large volume of data from our smartphone, tablet, PC. Times have changed data is growing rapidly and we need a system which can convert the data into information so that it can be used in different ways [7]. It is estimated that in coming years nearly 1.7 megabytes of information will be created by human beings [8]. So, we have good chance of getting good data from different sources whether it is social media data or any other type.

So in order to characterize that data and filter it according to user needs, we have focussed on using voice assistant which can interact with user and can give optimum results with the help of spoken words. With the help of handsfree voice assistants, characterization becomes simple and we can perform predictive analysis better and that too in real time. Our main focus in this paper is to analyze dataset using voice assistant.

In this research paper, we have experimented and evaluated our proposed approach with different data sets and using Alexa as a voice interface for our application and Lambda function to write our back end code which will be able to manage the data.

In this paper, back-end code logic is created in such a way that it can understand and able to read the data and give the results back to user. It can differentiate between attributes and data points and this application uses the basis of cloud computing. One big benefit of cloud computing is that it can be accessed from anywhere in the world and doesn't need any high computing machine to perform its work. We have tested our system on different data sets, such as 'prevalence of heart disease in US citizens', 're-admissions in hospitals' and Behavioural Risk factors and Stroke prevention' and recorded the results. Furthermore the system is explained in detail in Section 3.

1.1 Literature Review & Motivation

In some research it is being argued that Data Science is intricately intertwined with other important concepts as like of big data and machine learning [9]. There is a need to understand more about data science and how we can use it for our future. Right now we are looking at billions of data collected every day but cannot able to properly use it, in which we need to put more effort and techniques.

In other research, [10] there is a use of Bayesian non-parametric model was made to gather the data with the help of Artificial Intelligence and machine learning. Probabilistic approaches are very active area of research with wide-ranging impact beyond conventional pattern-recognition problems. And it is concluded that intelligence played a vital role in the system with probabilistic modeling remain central part of it.

It is being observed that with the changing of new era, some more languages have been introduced for Data Science. One of the languagee which is being considered for data science nowadays is Python, due to its abundance of open source libraries available and reducing the effort of writing long lines of codes and focussing more on logic. It is good for both supervised and unsupervised learning and provides a great ecosystem for the user. It is being used more than ever for machine learning and Natural language processing [11].

Researchers have also compared the cloud computing with other technologies and found that it will be a future of many existing systems. They have also proposed cloud technologies for creation of global Cloud exchange for trading services [12]. For that they need to extend their support for more cloud-based algorithms and allocating Virtual machines resources to meet the requirements. Large amount of Data centers are also required to store that much amount of data generated, but it will be a useful and more reliable investment for our future world.

In some other research, Alexa is being used for expression analysis. Alexa approach was followed by using human array to profile the expression of alternative mRNA isoforms in 5-fluourouracil (5-FU) sensitive and resistant colorectal cancer cell [**human**]. Then the results were compared with the Affymetrix's 'GeneChip® Human Exon 1.0 ST' array. Results were captured and discrepancies were showed using charts and figures.

Moreover, our motivation for this paper derived from Query-by-Voice using Alexa and SQL [13]. In the latest research, an end to end prototype system called EchoQuesry was introduced which allows Alexa to access data file and perform SQL queries on it. Alexa will send the response to HTTP endpoint and receive the response from it using Alexa Voice Service. Researchers were able to create a working model and tested it to get the results.

The system we are creating is independent of file and does not need specific type of data or anything. It can work on mostly all data files, does not matter what type data we are collecting whether it is social media or health data. We will further discuss about this in our later part of this paper.

Chapter 2

AWS & Alexa Skills Kit

2.1 Introduction to Alexa

Alexa is basically a virtual assistant developed by Amazon and is used is different devices such as Amazon Echo or Echo Dot smart speakers [14]. It is a voice assistant so user can activate it by using a wake word such as Alexa. It can perform different functions such as setting alarms, streaming podcasts, playing audiobooks, providing weather and, all other tasks which a virtual assistant can do with the help of Artificial intelligence and machine learning. Alexa uses Natural Language Understanding (NLU) to process the words spoken by the user and to make interaction more real and natural.

In figure 2.1 We can see how the whole process works and how user spoken words are converted from text to speech and invoke the skill. Our Database search tool also works in the same process as explained below.

1. First, User invokes the Alexa service by using wake word ’Alexa, open Database tool’ using Alexa powered smart device.

2. When Smart device receives the input from user, it simply passes on the input to Alexa voice service and, Alexa with the help of NLU breaks that input into words and try to understand what it actually meant with his input.

3. After Alexa understands the intent it sends the HTTP request to AWS Lambda ARN endpoint. AWS is Amazon Web Services which is used as an endpoint for Alexa to send and receive HTTP requests and responses, we will discuss about it further in this section later.

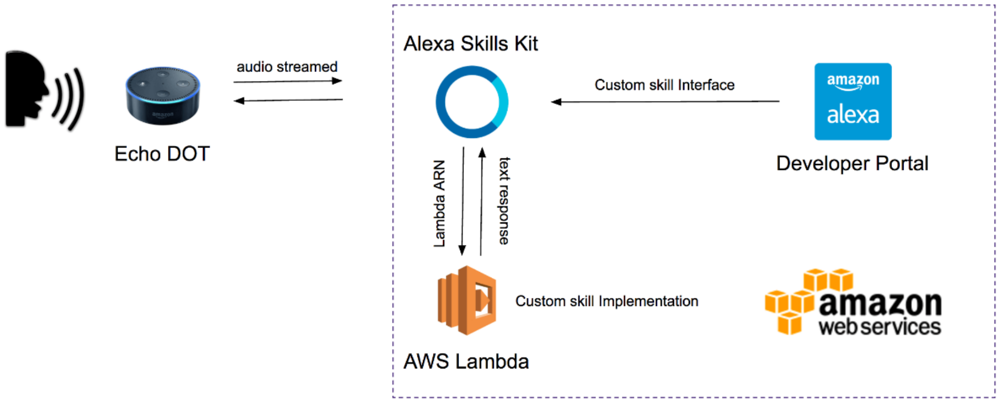


Figure 2.1: Alexa Voice Service [15]

4. After receiving HTTP requests from Alexa, it compiles the code written in some language and search for the called intent in the code. Then it prepares response for any method if it is called by user and sends it back to Alexa as an HTTP response.

5. After receiving response from AWS, Alexa uses that response to display the speech and replies back with 'Welcome to Database Search tool' to the user.

2.1.1 Alexa Voice Service

To understand user and its speech, Alexa uses intents, utterances, prompts and slots. Then it sends the computed speech to endpoint as JSON format requests [16]. In figure 2.2 we can see how internal process in Alexa works.

1. Firstly when user user invoke Alexa by using wake word, then user start by saying any of the utterances stored. Then, Alexa will match that utterance with utterances we build in the Alexa.

2. If any utterance matches with the utterances stored, then Alexa will invoke that intent and and make the JSON response with that intent name stored in it and send it as HTTP request to Lambda.

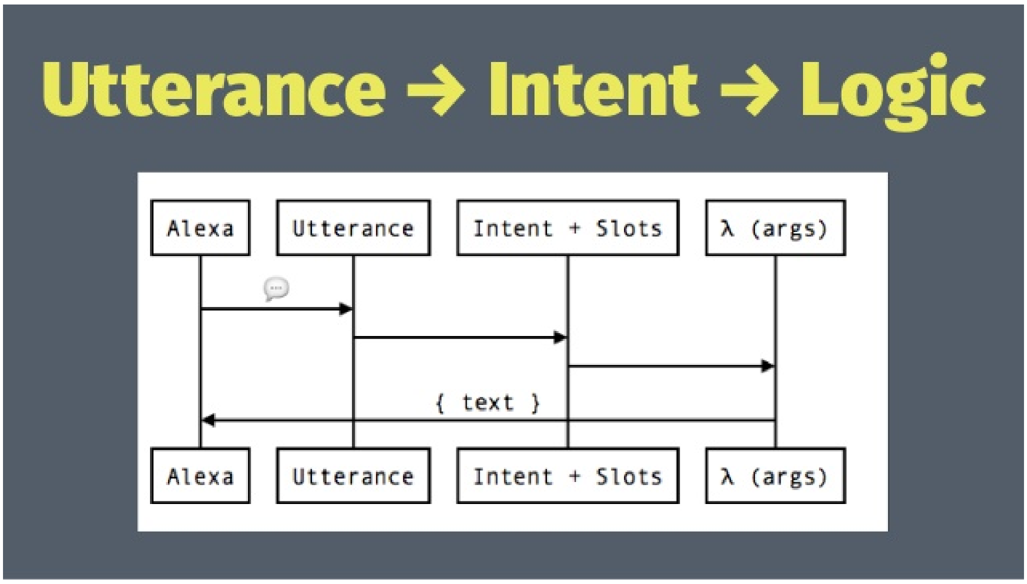

Utterance → Intent → Logic

Figure 2.2: Intents, Utterances, Slots & Lambda [17]

3. If that intent contains any slot then, its value is also sent in the request.
4. After receiving request if there is a prompt requirement by intent then the Alexa will prompt user, and user's response will be sent as a request to Lambda.

This is how it all works inside Alexa. The requests and responses are generated with the help of utterances, slots, intents and Lambda. We have created various intents for various purposes such as open_file, range, search, show_att, find_prob, save_data, clear_all, find_common, show_all_att. The purpose of each intent is described below.

- open_file: User will call this intent to open the data set he wants to work and perform other intent operations on. The file should be uploaded to S3 to use the file.
- range: This will display the range of the specific numeric attribute in the data file.
- search: This will search any keyword in the data file and let user know the occurrences of that word in the data file.
- show_att: This intent when invoked will display the specific attribute's distinct values or data points present in the data file.

- show_all_att: This intent is used to call all the attributes present in the data set.

- find_common: This is used to filter the data which user wants to from the data set. This requires user to give the attribute name and value which he want to filter from the data.

- find_prob: This is called by user when he wants to find the probability of the data which is filtered from the data set, and display the true cases in data file.

- save_data: This intent is used for storing all the filtered data into the cloud service S3 in a data file. User will provide with the name of data file.

- clear_all: This is simple intent which is invoked when user wants to clear the filtered data and wants to filter new data from data set.

All the responses sent to Lambda are in JSON format and varioud slot are defined to capture data from user. The Endpoint we are using is based on US-East server located in North Virginia.

2.2 AWS

Amazon Web Services are a group of cloud computing services provided by Amazon and are best for serverless computing and predictive analytics [18]. The services we are using in our application are Lambda and S3. Lambda is being used for our back-end code and S3 for storage service.

When user calls the intent to open the data file that is stored in S3, various process takes place between AWS, Lambda and S3 to access the data file. In figure 2.3, firstly, request is passed from Amazon CloudFront to Amazon gateway API which then passes the requests to Lambda function. Lambda function is used to write our code in Python language, it is described in later part of this section. Then lambda calls the S3 API and access the data file stored in S3 bucket. And the response is sent back to Lambda and this is how data file is accessed to work with by Alexa.

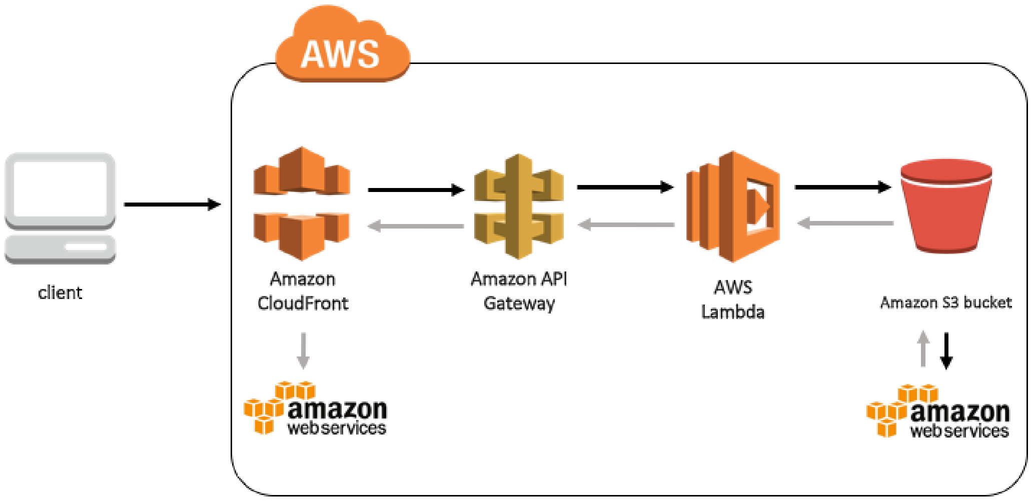


Figure 2.3: AWS architecture [19]

2.2.1 Lambda

Lambda is one of the service in AWS and used in our application to write our back-end logic. It acts as a brain to our application and all the computations and operations on data file are done with the help of this service. It is a serverless cloud service which give it a big advantage over other services [20]. It is independent of any location or system and can be accessed from anywhere in the World.

The application's code is written in Python 2.7, and different methods are used to perform different functions. How this works is when user invoke any intent using his utterance then that intent calls the Lambda code and each of the utterance have some method to call which will perform some operation. So, what it means that different intents will call different methods in the code and perform some unique function depends on the method. We have created different methods or functions such as open_data(), search_data, ranges_, show_all_att(), show_att(), find_common(), save_data(), find_prob(), clear_all(). As we discussed above in intents, these methods are back-end code for those intents and gives the ability to perform the operations to intents.

The one core feature of Python is Extensive support of libraries available to use [21]. We are using some libraries to read and write data in S3 such as csv, re, urlib2, boto3, izip, etc. and to do searching and calculations.

2.2.2 S3

Simple Storage Service is also a cloud based service. It is used in our application to store data file, which we are uploading through a HTML page interface. We have to create a bucket to store any file in S3. We are uploading our file into a bucket named 'lawishbucket' through one page interface which is created using JavaScript and HTML. Once the file is uploaded, Alexa can access that file through its S3 API. If there is no file in the S3 which user wants Alexa to access, then Alexa will send user a response saying 'File not found, please try again!'. So, the file should be present in order for Alexa to access it.

Chapter 3

Methodology

3.1 Framework Architecture

In this paper, we are using Artificial Intelligence with the help of speech recognition using Alexa to perform different operations and statistics calculations on data 3.1.

We can use any data file which is a comma separated values (.csv) file. There are several processes which took place in this project which we will go through in detail step by step.

1. System architecture
2. Storing data in S3
3. AWS - Cloud Computing Services
4. Speech Recognition
5. Amazon developer console
6. Dialog Delegates
7. Testing skill - HTTP Requests and Responses

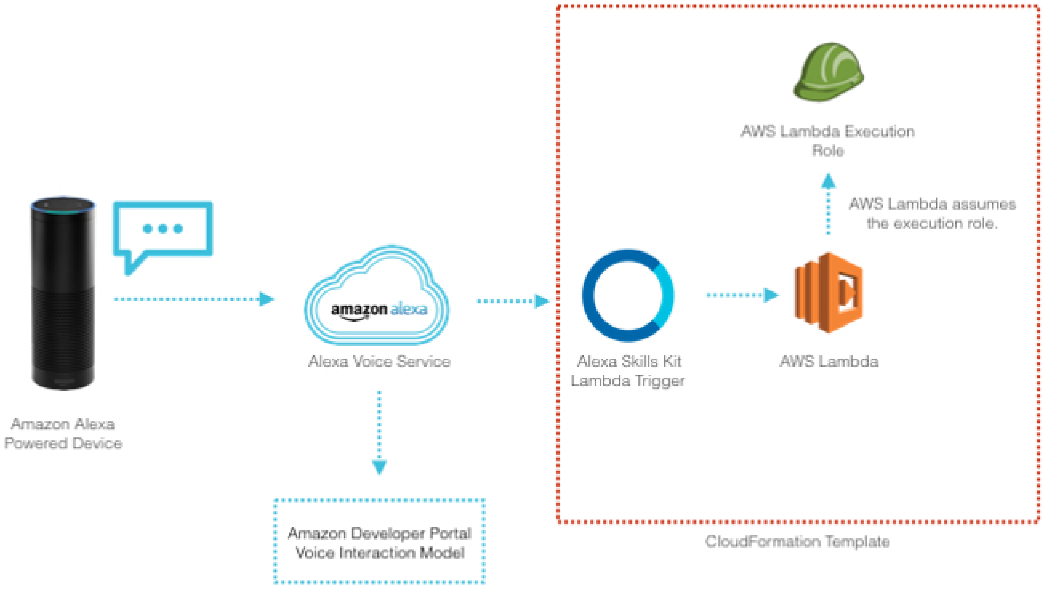

AWS Lambda Execution Role
AWS Lambda assumes the execution role.
amazon alexa
Alexa Voice Service
Alexa Skills Kit Lambda Trigger
AWS Lambda
Amazon Alexa Powered Device
Amazon Developer Portal Voice Interaction Model
CloudFormation Template

Figure 3.1: Framework architecture [22]

3.1.1 System architecture

The system architecture works and process data in real time. It consists of several processes and components, which are as follows.

- Alexa powered device
- Lambda Execution role in IAM
- Skills kit
- Amazon Developer portal

To make skills perform some particular operations and decisions we need to call them and access them using some sort of device which can listen to users speaking and send commands forward accordingly [23]. So, this is the first step through which system get the raw input from the user and process it further using speech recognition 3.2. Currently, Alexa is not available for any other device except these.

amazon
Brand
Try "Alexa, show more"
Amazon, Kindle, Echo, Alexa, Dash, Fire and all related logos are trademarks of Amazon.com, Inc. or its affiliates.

Figure 3.2: Alexa Devices [24]

A Lambda function requires an execution role created in IAM that provides the function with the necessary permissions to run. Every function is given some execution role and permission in IAM that are required and level of permission depends on computations that function is doing. Lambda is a serverless, open-source function written using Node.js and is used widely for different technologies as well.

We will be creating our skills using Alexa skills kit. So, we build our logic in the Alexa skill kit, publish it and use it with the Lambda function to make it work with Alexa devices to get the desired response [25]. All the code we create is saved in the cloud using AWS lmabda service.

Amazon Developer portal is the place where we can build intents, entitites for our skill, and we can set interfaces, endpoints and other configuration settings for our skills. We can have look at our JSON requests and responses as well in developer console.

3.1.2 Storing data in S3

We will be hosting our file on Simple Storage Service (S3) by Amazon. S3 is based on the concept of object storage using cloud [26]. We can create a bucket in S3 and can save our data in that, so that it can be accessible directly through http requests. Basically, buckets are similar to folders in the storage 3.3. There are various reasons for hosting file on S3 as follows,

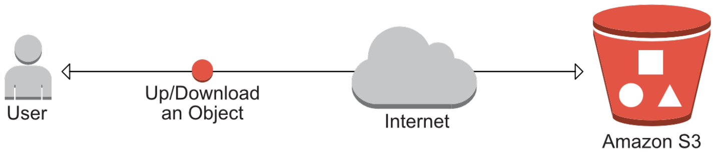


Figure 3.3: S3 Storage [27]

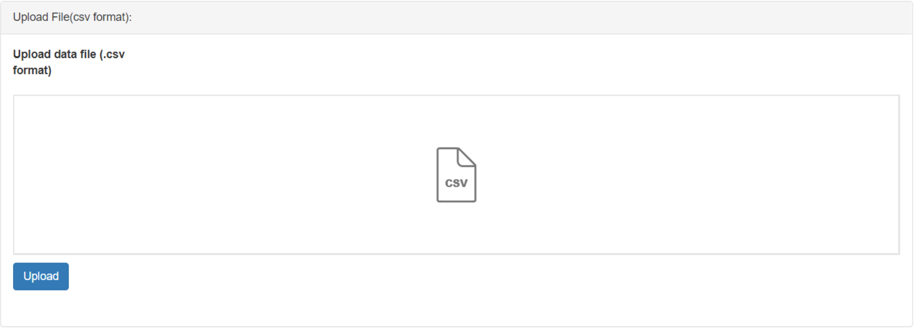
- **Performance:** S3 aims to provide scalability, low latency with 99.9% durability and up to 99.99% availability in servers. It creates and stores copies of all S3 objects throughout more than one systems. This means that our data is accessible when needed and protected against failures, errors, and threats.

- **Secure Storage:** It stores our data and secures it from unauthorized access with encryption features and access management tools. It also prevents it from irregular access to our data, ensuring no restricted access can have a peak. To access our data we have to define a policy with X-AMZ algorithm, credentials and signature values. Access Id and secret key is required every time we need to provide anybody with access to our data, which can be generated by a person who has admin rights to bucket.

- **Accessibility:** One huge reason to store file on cloud is the accessibility, we can access the file anywhere, anytime through Alexa. We don't need to be on a specific machine to access it, we can just say "open {filename}" to Alexa running on any machine, and it will open our file and we are ready to go.

- **Data Transfer Rates:** S3 can handle large amount of transferring of data easily and faster due to the fast servers speed and will reduce time taken to read and parse data file for Alexa.

As we know, Alexa does not allow any uploading functionality. So, to overcome that we are using a HTML page interface to upload files in S3 database 3.4. User can upload any CSV data file to work with, using this interface.


Upload File(csv format):

Upload data file (.csv format)

Figure 3.4: File Uploading Interface

After uploading the data file into S3, with the help of S3 API, we can access the file through Alexa and will be able to work with that file.

3.1.3 AWS - Cloud Computing Services

AWS is a cloud based service provided by Amazon which gives developers to develop various types of applications using different services 3.5. ”Amazon Web Services offers a set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security and enterprise applications” [28].

In our project we will be using some of the AWS services, such as Lambda for writing our code, S3 API to store and retrieve data, Alexa skills kit API to enable speech configuration in our application. Using AWS Lambda has some benefits which we will discuss later in this section. All of these services in our project operates and compute results in real time. We will further discuss about these services in brief, as follows:

Figure 3.5: AWS as a Cloud Service [29]

Lambda

AWS Lambda is a serverless compute service that runs our code in response to events and manages the underlying compute resources for us. So, basically it provides us with all the resources and computing power we need and we just need to make our logic and code to make a custom application.

We can use AWS Lambda to extend other AWS services with our custom code, or create our own back-end services that operate at AWS scale, performance, and security [30]. AWS Lambda can run the code in response to multiple events, such as HTTP requests via Amazon API Gateway, modifications to objects in Amazon S3 buckets, etc. This eliminates the problem of any servers or domain needs.

Lambda runs our code on high-availability compute infrastructure and performs all the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling [31], code and security patch deployment, and code monitoring and logging.

We are writing our code in Lambda using Python language. Python is a high level programming language which is normally known as general purpose programming language. Python is idle for working with data and is popularly used in Data Science field 3.6.

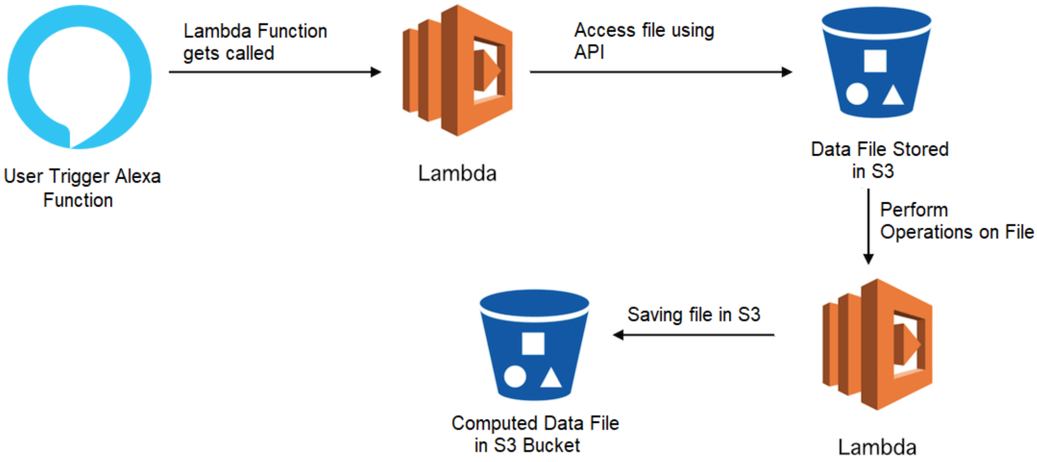

User Trigger Alexa Function
Lambda Function gets called
Lambda
Access file using API
Data File Stored in S3
Perform Operations on File
Lambda
Saving file in S3
Computed Data File in S3 Bucket

Figure 3.6: Project Architecture

3.1.4 Speech Recognition

Speech recognition refers to the ability of machine to recognize texts or words in spoken language. We are using Alexa for our speech recognition. Alexa uses back-end technologies and APIs to detect spoken sound and recognize them as words 3.7. We can access Alexa by using some phrases like "Alexa, open database tool" [32].

Figure 3.7: Alexa Speech Processing [33]

Alexa uses Natural language processing (NLP) to process the speech and then use those words to perform functions using Lambda. There are various reason for using Alexa as a voice service, described below [34].

- **Fast:** Alexa is really fast when it comes to speed. If conversions happens in milliseconds then conversation feels natural, that is why user feels more comfortable in using Alexa. Alexa uses cloud computing to break the audio into words which helps machine to act on it instantly. This helps Alexa making long and reliable conversations with user.

- **Educated Decisions:** We know that there can be more than one meaning to some texts, so to deal with it Alexa uses back-end technologies to get the exact meaning of texts spoken. It analyses the whole sentence rather than a word to determine what is user meant by his speech.

- **Improvised Learning:** Alexa uses speech recognition to develop its learning with time, the more user interact with Alexa, the more Alexa becomes good with its speech. Natural Language Understanding will improve itself over time and make good decisions based on its learning intelligence.

  We are using different speech dialogs in our project. To open our skill we have to call Alexa to open it and then Alexa will interact with Lambda to run it. For an instance, we will say - "Alexa open Database tool" and Alexa will run the skill and ask for the file name we want to open and work on.

3.1.5 Amazon developer console

Amazon developer console is a cloud based platform which allows user to use various services and APIs to make an application. We are developing our skill using this console [35]. We are using different modules in our application such as interaction model, endpoint, testing, analytics 3.8.

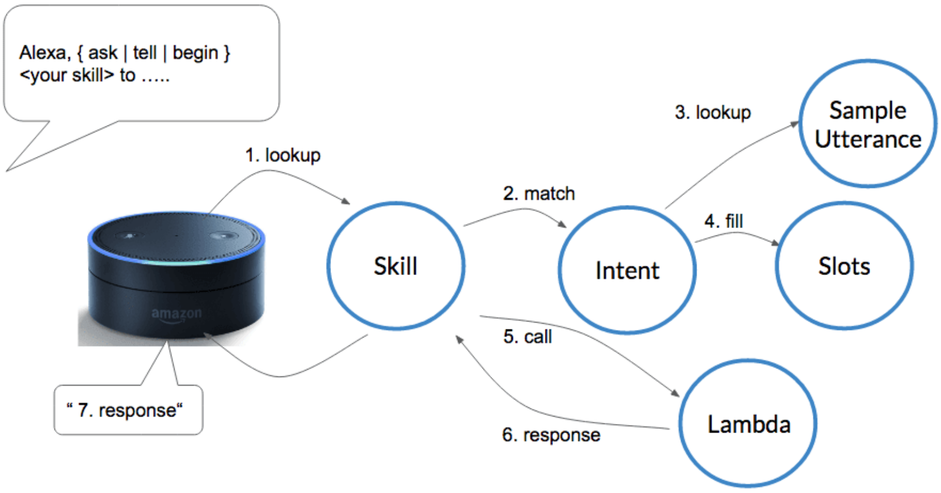


Figure 3.8: Developer Console Architecture [36]

Interaction Model

Interaction model refers to the custom model to interact with Alexa to perform required operations [37]. To build an interaction model, we need to create intents, utterances and slots and their types. After specifying all these we can complete our model and connect it to the Alexa skill set. These processes are described below in more details.

- **Intents:** Intents are basically actions or responses that are used to fulfill the requests made by user. It is used to capture data from user and can use that data to perform other operations. For instance, user can book a flight ticket to somewhere by providing city name. These are made up of Utterances and slots.

  There are two types intents we are using in our application, built in and custom intents [38]. Built in intents are predefined intents to perform specific operations such as Help, Stop, Cancel, etc. We can't change them but can only modify their utterances. The other type of intents we are using in our application is custom intents. These intents are made as per the application requirements and are used for performing operations for specific applications. User can create them and have full freedom to do whatever operations he wants them to perform. We have created several custom intents in our application to perform various operations.

- **Utterances:** Utterances can be defined as sentences which are used to invoke the skill or method to perform some operation. It can consist of words or phrases which Alexa can expect from user to speak to invoke it. Some examples of utterances are - ”Alexa, Book my flight to San Francisco”, in which San Francisco is a data given by user and will be used in slot [39]. With the help of Utterances only system gets to know what intents needs to run and what output will be sent to Alexa.

  We are using various utterances in our application. For an instance to show the attributes in the data file, user will say ’show att’ and Alexa will display all the attributes in the data file or to display any specific attribute then we need to say ’show ¡attribute name¿’. This is how we are using utterances in our applications. We are also using utterances to capture slots values to operate on data file. We will discuss about slots further in detail below.

- **Slots:** Slots are defined as a type of variables used by Alexa interface to store some data or values, depending on the user requirements. There are different types of slots available which we can use [40]. We are only using SearchQuery in our application due to the fact that our application is based on searching in a file and it can be unpredictable data so this is the best option we get. There are other slot types also used for other purposes such as AMAZON.Actor, AMAZON.Airline, AMAZON.Airport, AMAZON.City, and so on.

Endpoint

When the user sends the post requests to the endpoint and endpoint sends the response then only the user is able to interact with the skill. The request is formatted in JSON and is used to create logic or do function on the basis of it. Without Endpoint we can’t use our skill as there will be no place to send the requests and receive the responses from [41]. There are two types of Endpoints which Alexa supports, that are AWS Lambda ARN and HTTPS. HTTPS endpoint is managed by user’s web service and we are using AWS Lambda for our application as it is easy to manage and things get done.

We first create the Lambda function in Amazon Web Services portal, then we take the URL and paste it into the Endpoint section. As we discussed above about Lambda, we write our code in Lambda, so the post request go to the Lambda and then it sends the response back to Alexa. This is how Endpoint system works.

Testing and Analytics

When we finish developing our skill, one of the most important thing is to test the skill, is it working as expected or not. That is what testing module does, it helps us test our skill and also let us know the errors in our code if we have any. This is like a small display interface module, which shows us the results we are getting and how will it look and displayed on any Alexa device [42]. We can select different display interfaces to choose from. Testing is important as it puts our application to test and tells us about any potential warning or irregular results if we are getting any. We can also change voice or tone of Alexa and JSON input according to our needs.

Analytics on the other hand tells us about the insights of our project such as sessions, utterances, intents, Interaction Path, Retention, Skill Activation. It tells us total successful sessions or how many times session is opened or skill is invoked and number of errors we got during our skill invocation. It helps us understand our skill better and number of failures and successes we are getting with our application.

3.1.6 Dialog Delegates

Many times in real life when we have a conversation with someone, ”We would ask questions to gather information. We are gathering a set of information and interpreting that data” [43]. In the same way Alexa use Dialog Directives for multi-turn conversations between Alexa and user or in other words it acts as a type of prompt [44]. It can be used to fulfill the information needed from the user. To use this we need to create dialog model, which will set a order in which the information will be stored and take care of some other necessary things such as asking for prompts, intent confirmation, setting slot values, confirming user response and so on. There are four types of dialogs we are allowed to use that are Delegate Directive, ElicitSlot Directive, ConfirmSlot Directive, ConfirmIntent Directive [45]. We are all of them except for the ConfirmIntent in our application. 3.9 shows us the example of Dialog directive, in which Alexa is asking for information from the user using prompts, then the user will provide the information and it will move on to next prompt. Directives used in our project are further described below.

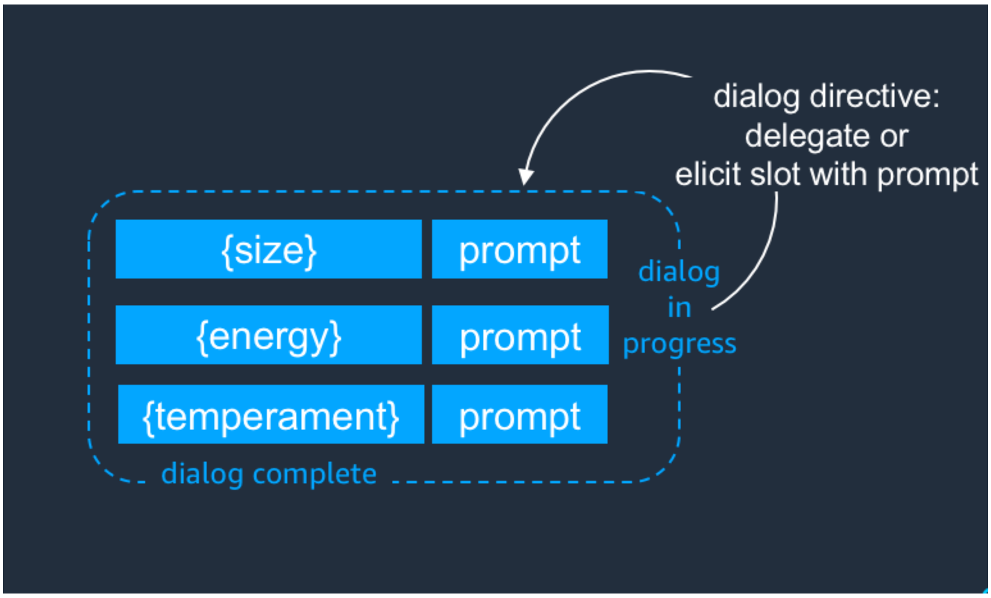


Figure 3.9: Dialog Directive Example [46]

Delegate Directive

Delegate Directive is used to send the command to the Alexa to handle the next turn with the user. It is the most basic of all other dialogs and only used for next turns and the next step is determined by dialog model. It has some parameters such as type and updatedIntent. type is a string and updatedIntent is an object, both are necessary to pass the information and make it work.

ElicitSlot Directive

ElicitSlot Directive is used to send a command to Alexa to ask user for a value of specific slot. We do have to specify which slot and intent we are updating and it will store the value by user in that slot. There is a particular syntax which needs to be followed along with some specific parameters. Some parameters are type, slotToElicit, updatedIntent, these all are strings except for updatedIntent which is an object. We use updatedIntent object to specify the slot and intent which we are modifying and also set the confirmation status.

ConfirmSlot Directive

This is one of more advance directive from other and most useful of them. It can act as confirmation for certain slot in an intent as well as can be used to modify the slot value. It provides a prompt in an OutputSpeech object and specify the name of the slot in slotToConfirm property. It has 2 string parameters and one object parameter which are used to modify any value or for confirmation. It can perform both the functions of Delegate and ElicitSlot dialogs.

3.1.7 HTTP Requests and Responses

HTTP requests and responses are used to send or receive data in Alexa. There are certain parameters by which these data is sent and these are necessary for the proper flow of data. Some parameters of which are mandatory and some are optional. 3.10.

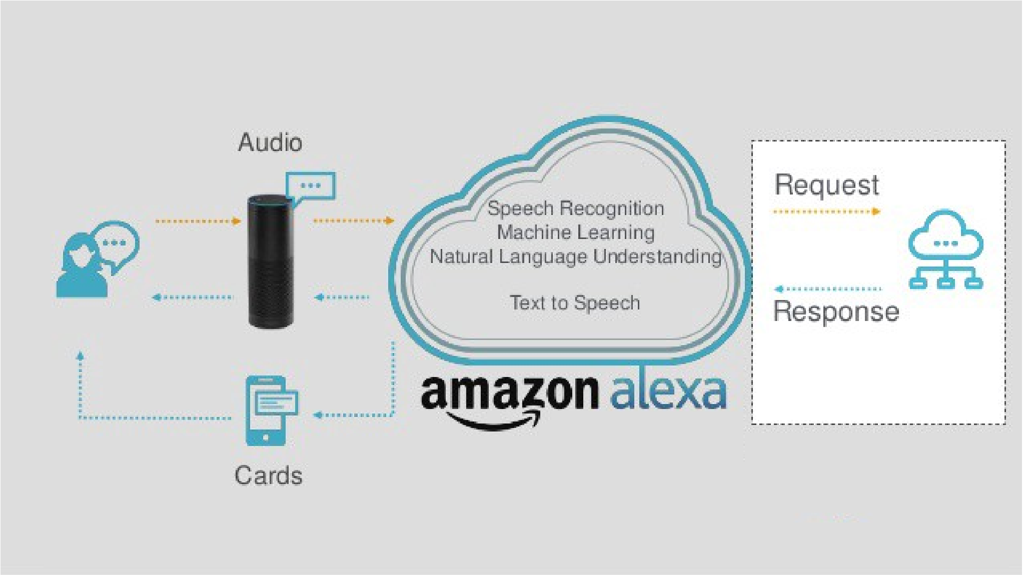


Figure 3.10: Request and Response [47]

Request body parameters used are - version, session, context, request. All the requests and response should be in a JSON format to work [48]. These are further explained below.

- **Version:** This is used to specify the version number Alexa is currently at and is always sets to ”1.0”

- **Session:** Session object is used in various standard request types such as CanFulfillIntentReqeuest, LaunchRequest, IntentRequest, and SessionEndedRequest. There are different values passed in session object, mostly strings, such as sessionId, attributes, application, user. This helps Alexa in getting user information and status and also if user wants to end the session after calling the method or not. Session object is not used in AudioPlayer and PlaybackController.

- **Context:** This object provides the current state of the Alexa and device to our skill service. It includes all the requests which are present in Session object also. It duplicates some objects and send it to the service. Values passed are System and AudioPlayer.

- **Request:** This provides details about the user's request. There are several different request types available, Standard Requests such as CanFulfillIntentRequest, LaunchRequest, IntentRequest, SessionEndedRequest, AudioPlayer Requests, PlaybackController Requests, Display.ElementSelected Requests, GadgetController Requests, GameEngine Requests.

Responses are the returns we get after sending our requests to the service. There are some parameters of response as well such as version, sessionAttributes, response. In these, the version remains the same "1.0", sessionAttributes is changed if the the session changes, it contains one string value and one object, which changes accordingly. Response object contains various values which can be strings, object, boolean or array depending on value. It provides the output to user and also tell the system if it needs to end the session or not. Different values in Response are outputSpeech, card, reprompt, shouldEndSession, directives.

Chapter 4

Results & Analysis

For this application, we performed our results on dataset about prevalence of diabetes among different age group of people from different territories. All the commands were given to Alexa by voice including opening database, filtering rows, finding probability and displaying attributes. The dataset contains more than 20k rows. It contains different attributes such as geo, age group, diabetes, characteristics and so on.

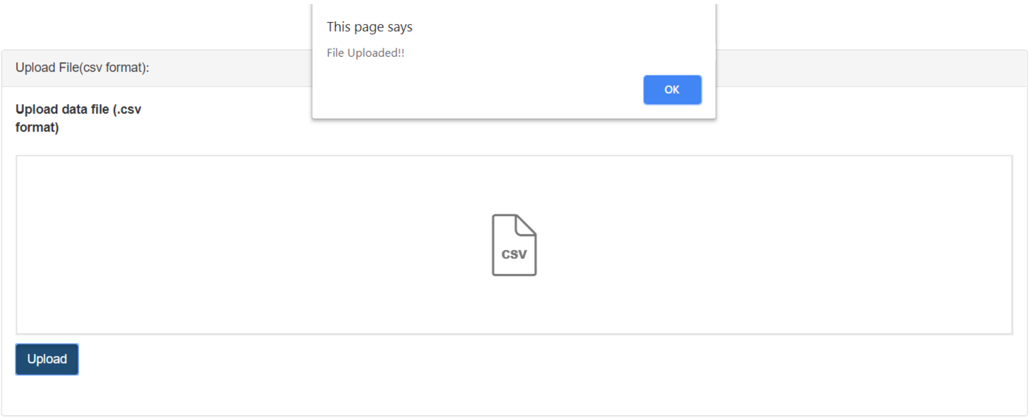
4.1 Technological Requirements

This project uses Python as a main language for computations, but HTML and JavaScript are used for making upload interface. Lambda function is used as an environment provided by Amazon to write our code and S3 is used for storing our data.

- Backend
  - Python
  - HTML & JavaScript
  - S3
  - Lambda
- Libraries
  - OrderedDict
  - izip
  - csv
  - boto3
  - urllib2

4.2 Uploading Dataset to S3

Firstly, we are uploading a dataset to S3 on which we want to work on and for that we are using a simple HTML based interface as shown in 4.1. It is a drag & drop interface and is compatible for any CSV file.



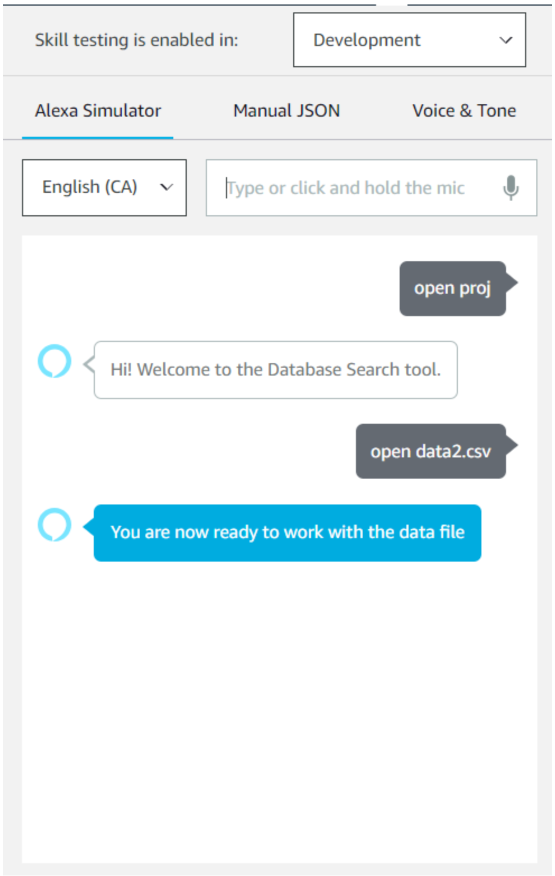
This page says

File Uploaded!!

Figure 4.1: Uploading File

4.3 Data Filter

For running our skill we are currently using Alexa Simulator, which is same like using any Alexa enabled device and give the exact same results. So, 4.2 in we are giving command to open our project that is ”open proj” and Alexa respond with Hello response. After opening the skill, we are asking Alexa to open the dataset and it replies with ready to work with it.



open proj

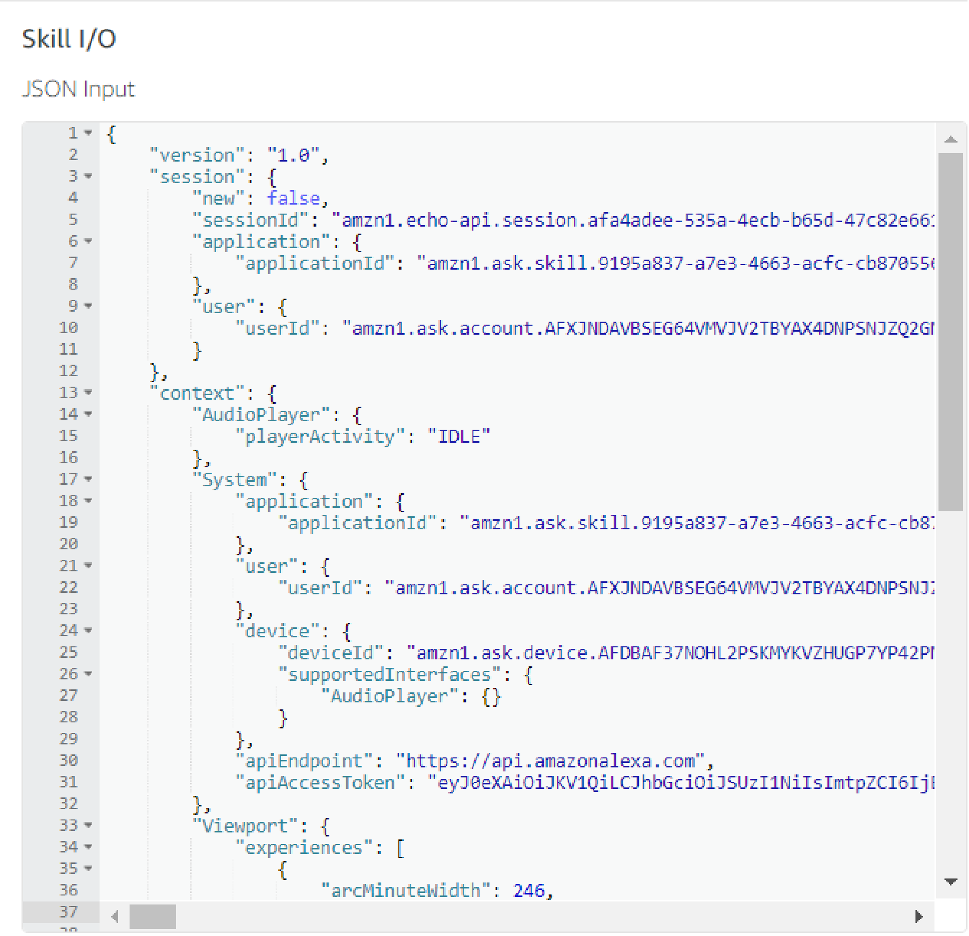
Hi! Welcome to the Database Search tool.

open data2.csv

You are now ready to work with the data file

Figure 4.2: Opening skill

- This is how JSON request looks like for opening a data file 4.3.



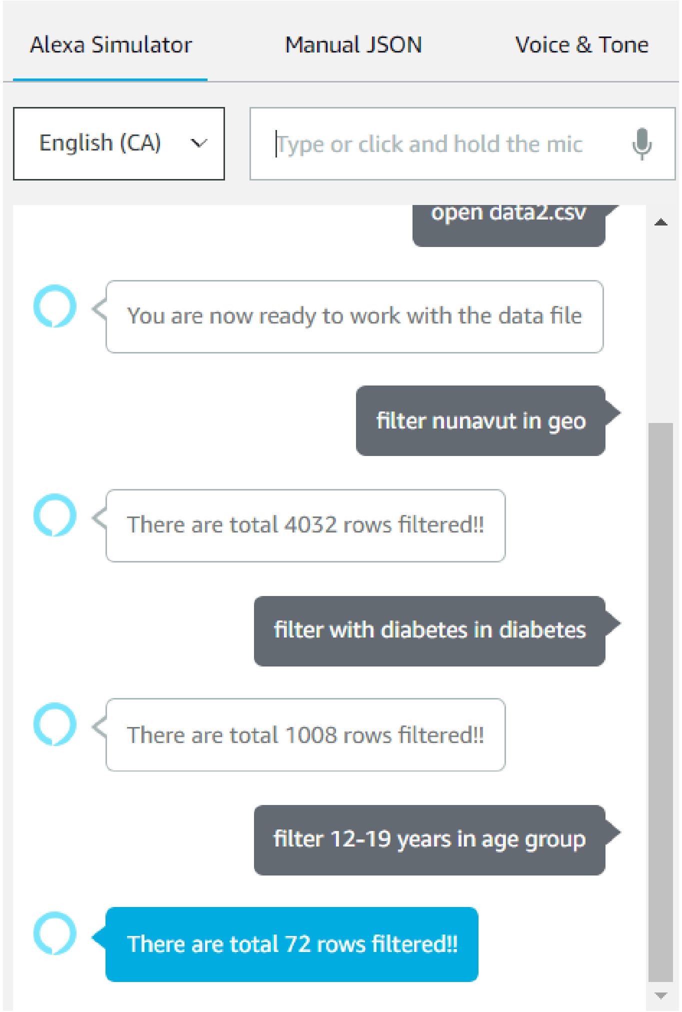
Skill I/O

JSON Input

```
{
    "version": "1.0",
    "session": {
        "new": false,
        "sessionId": "amzn1.echo-api.session.afa4adee-535a-4ecb-b65d-47c82e66
        "application": {
            "applicationId": "amzn1.ask.skill.9195a837-a7e3-4663-acfc-cb87055
        },
        "user": {
            "userId": "amzn1.ask.account.AFXJNDAVBSEG64VMVJV2TBYAX4DNPSNJZQ2G
        }
    },
    "context": {
        "AudioPlayer": {
            "playerActivity": "IDLE"
        },
        "System": {
            "application": {
                "applicationId": "amzn1.ask.skill.9195a837-a7e3-4663-acfc-cb8
            },
            "user": {
                "userId": "amzn1.ask.account.AFXJNDAVBSEG64VMVJV2TBYAX4DNPSNJ
            },
            "device": {
                "deviceId": "amzn1.ask.device.AFDBAF37NOHL2PSKMYKVZHUGP7YP42P
                "supportedInterfaces": {
                    "AudioPlayer": {}
                }
            },
            "apiEndpoint": "https://api.amazonalexa.com",
            "apiAccessToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Ij
        },
        "Viewport": {
            "experiences": [
                {
                    "arcMinuteWidth": 246,
```

Figure 4.3: JSON Request

- In figure 4.4, this is how we will keep on filtering data until user wants the desired data. The system will tell us how many rows are filtered throughout the process and if we enter any invalid data, then it will give response as ”Invalid input”.



open data2.csv

You are now ready to work with the data file

filter nunavut in geo

There are total 4032 rows filtered!!

filter with diabetes in diabetes

There are total 1008 rows filtered!!

filter 12-19 years in age group

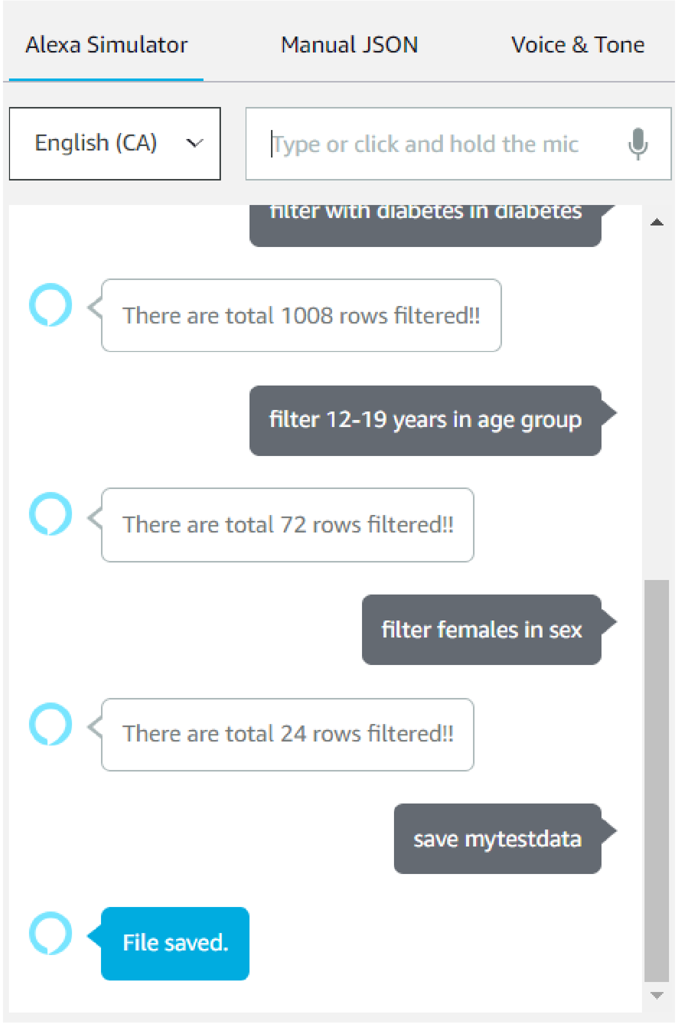
There are total 72 rows filtered!!

Figure 4.4: Refining Data

- So we are finding females who have Diabetes in the age-group of 12-19 years and are from Nunavut.

4.4 Saving Data

So we got 24 rows filtered from all the data after applying various filters. In figure 4.5 we are giving ”save ¡file-name¿” command to Alexa to save the data in S3 in CSV format. Alexa respond back with ”file saved” response, which means file is saved in database.



filter with diabetes in diabetes

There are total 1008 rows filtered!!

filter 12-19 years in age group

There are total 72 rows filtered!!

filter females in sex

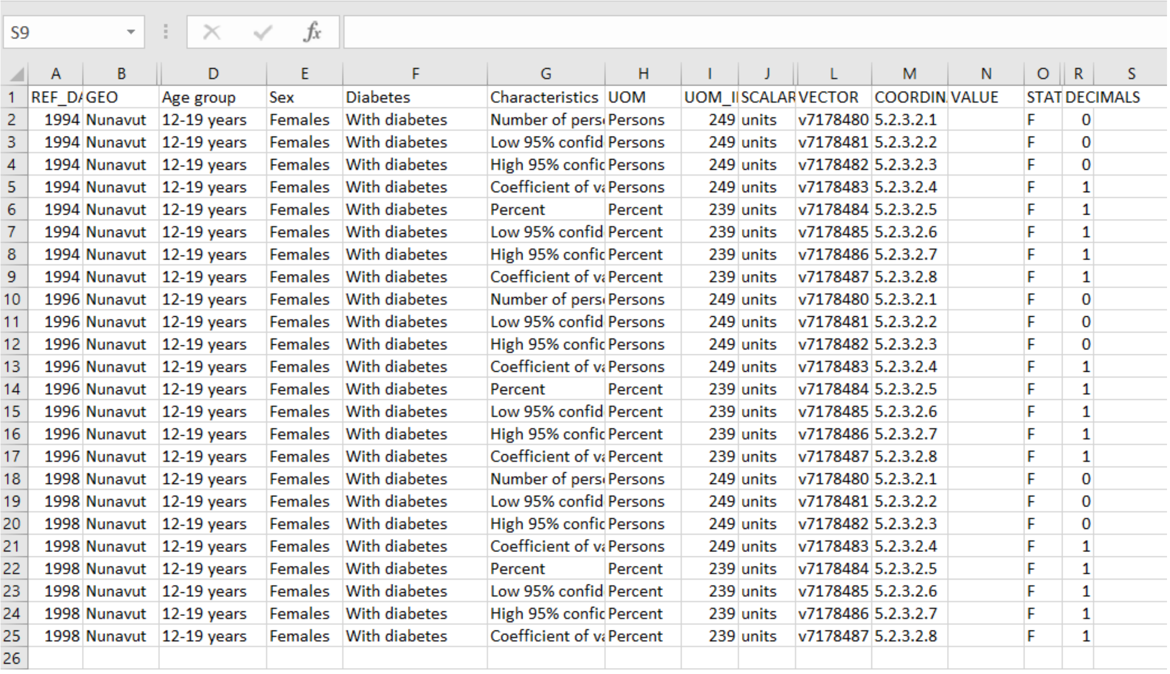
There are total 24 rows filtered!!

save mytestdata

File saved.

Figure 4.5: Saving Data

- This is 4.6 how our new data file looks like after we just saved, and provide us with required results.

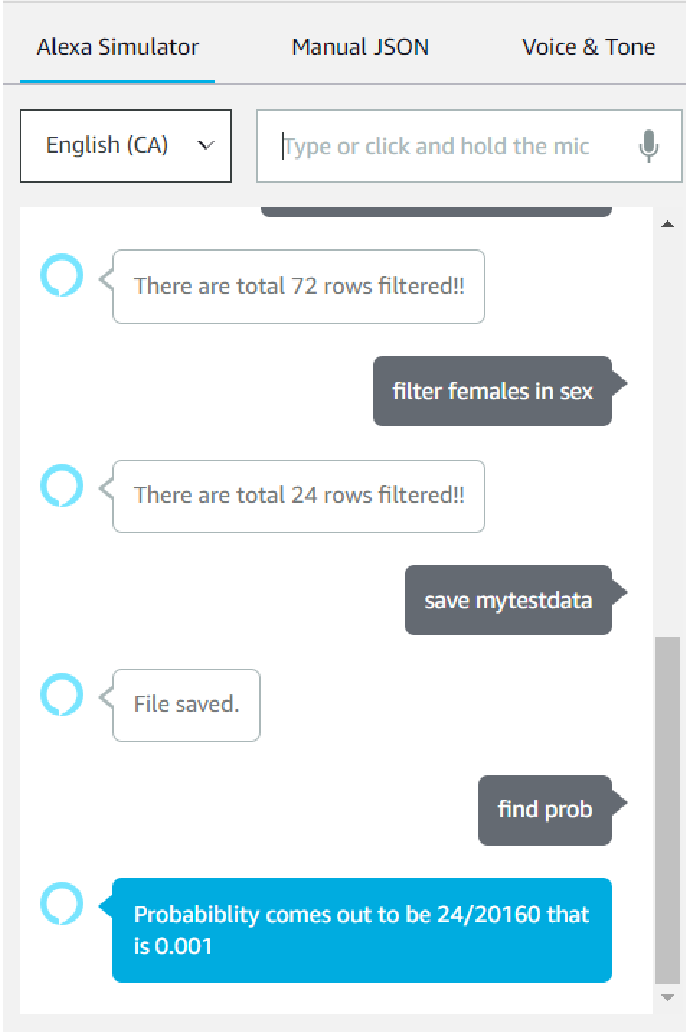


	A	B	D	E	F	G	H	I	J	L	M	N	O	R	S
1	REF_D	GEO	Age group	Sex	Diabetes	Characteristics	UOM	UOM_I	SCALAR	VECTOR	COORDIN	VALUE	STAT	DECIMALS	
2	1994	Nunavut	12-19 years	Females	With diabetes	Number of pers	Persons	249	units	v7178480	5.2.3.2.1		F	0	
3	1994	Nunavut	12-19 years	Females	With diabetes	Low 95% confid	Persons	249	units	v7178481	5.2.3.2.2		F	0	
4	1994	Nunavut	12-19 years	Females	With diabetes	High 95% confic	Persons	249	units	v7178482	5.2.3.2.3		F	0	
5	1994	Nunavut	12-19 years	Females	With diabetes	Coefficient of v	Persons	249	units	v7178483	5.2.3.2.4		F	1	
6	1994	Nunavut	12-19 years	Females	With diabetes	Percent	Percent	239	units	v7178484	5.2.3.2.5		F	1	
7	1994	Nunavut	12-19 years	Females	With diabetes	Low 95% confid	Percent	239	units	v7178485	5.2.3.2.6		F	1	
8	1994	Nunavut	12-19 years	Females	With diabetes	High 95% confic	Percent	239	units	v7178486	5.2.3.2.7		F	1	
9	1994	Nunavut	12-19 years	Females	With diabetes	Coefficient of v	Percent	239	units	v7178487	5.2.3.2.8		F	1	
10	1996	Nunavut	12-19 years	Females	With diabetes	Number of pers	Persons	249	units	v7178480	5.2.3.2.1		F	0	
11	1996	Nunavut	12-19 years	Females	With diabetes	Low 95% confid	Persons	249	units	v7178481	5.2.3.2.2		F	0	
12	1996	Nunavut	12-19 years	Females	With diabetes	High 95% confic	Persons	249	units	v7178482	5.2.3.2.3		F	0	
13	1996	Nunavut	12-19 years	Females	With diabetes	Coefficient of v	Persons	249	units	v7178483	5.2.3.2.4		F	1	
14	1996	Nunavut	12-19 years	Females	With diabetes	Percent	Percent	239	units	v7178484	5.2.3.2.5		F	1	
15	1996	Nunavut	12-19 years	Females	With diabetes	Low 95% confid	Percent	239	units	v7178485	5.2.3.2.6		F	1	
16	1996	Nunavut	12-19 years	Females	With diabetes	High 95% confic	Percent	239	units	v7178486	5.2.3.2.7		F	1	
17	1996	Nunavut	12-19 years	Females	With diabetes	Coefficient of v	Percent	239	units	v7178487	5.2.3.2.8		F	1	
18	1998	Nunavut	12-19 years	Females	With diabetes	Number of pers	Persons	249	units	v7178480	5.2.3.2.1		F	0	
19	1998	Nunavut	12-19 years	Females	With diabetes	Low 95% confid	Persons	249	units	v7178481	5.2.3.2.2		F	0	
20	1998	Nunavut	12-19 years	Females	With diabetes	High 95% confic	Persons	249	units	v7178482	5.2.3.2.3		F	0	
21	1998	Nunavut	12-19 years	Females	With diabetes	Coefficient of v	Persons	249	units	v7178483	5.2.3.2.4		F	1	
22	1998	Nunavut	12-19 years	Females	With diabetes	Percent	Percent	239	units	v7178484	5.2.3.2.5		F	1	
23	1998	Nunavut	12-19 years	Females	With diabetes	Low 95% confid	Percent	239	units	v7178485	5.2.3.2.6		F	1	
24	1998	Nunavut	12-19 years	Females	With diabetes	High 95% confic	Percent	239	units	v7178486	5.2.3.2.7		F	1	
25	1998	Nunavut	12-19 years	Females	With diabetes	Coefficient of v	Percent	239	units	v7178487	5.2.3.2.8		F	1	
26															

Figure 4.6: New data-file

4.5 Probability

Probability of the data we refined can be calculated by giving ”find prob” command to Alexa 4.7, which comes out to be 0.001 for these results.



There are total 72 rows filtered!!

filter females in sex

There are total 24 rows filtered!!

save mytestdata

File saved.

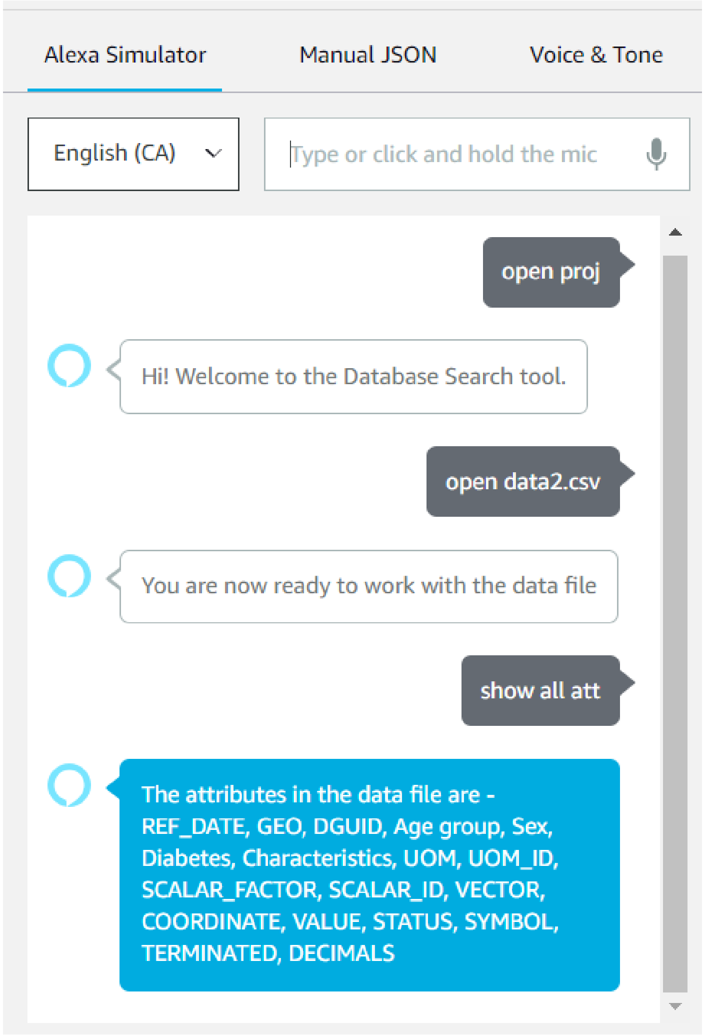
find prob

Probabiblity comes out to be 24/20160 that is 0.001

Figure 4.7: Probability

4.6 Display Attributes

User can see which attributes are present in the data-set and can also see for the specific attribute's values. In figure 4.8, this is how Alexa will display all the attributes using "show all att" command and in figure 4.9, for specific attribute and its values.



open proj

Hi! Welcome to the Database Search tool.

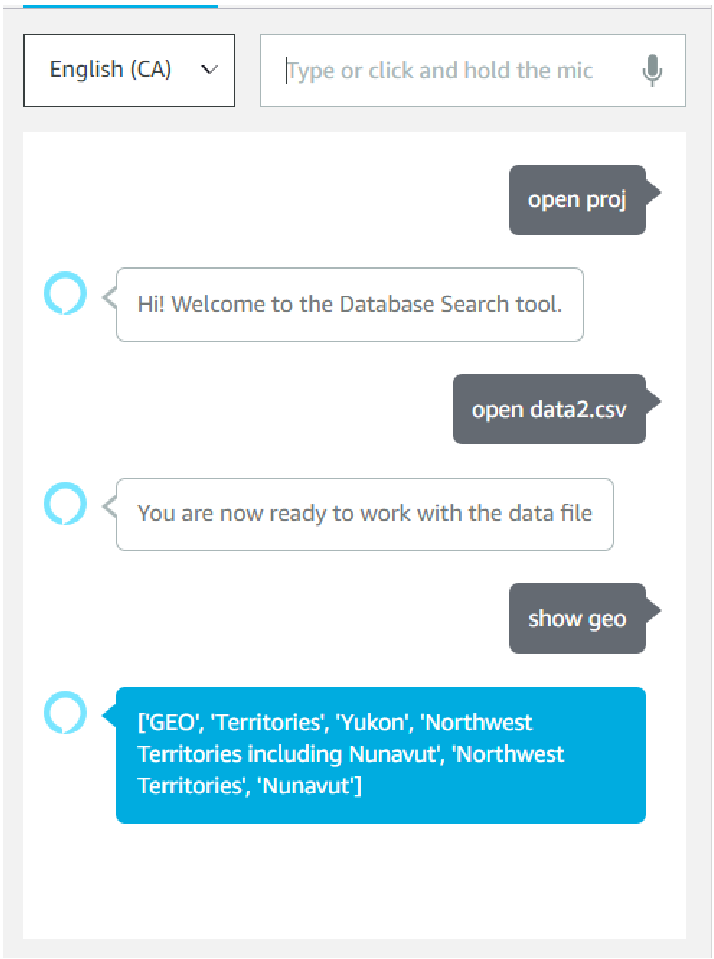
open data2.csv

You are now ready to work with the data file

show all att

The attributes in the data file are - REF_DATE, GEO, DGUID, Age group, Sex, Diabetes, Characteristics, UOM, UOM_ID, SCALAR_FACTOR, SCALAR_ID, VECTOR, COORDINATE, VALUE, STATUS, SYMBOL, TERMINATED, DECIMALS

Figure 4.8: Display (a)



open proj

Hi! Welcome to the Database Search tool.

open data2.csv

You are now ready to work with the data file

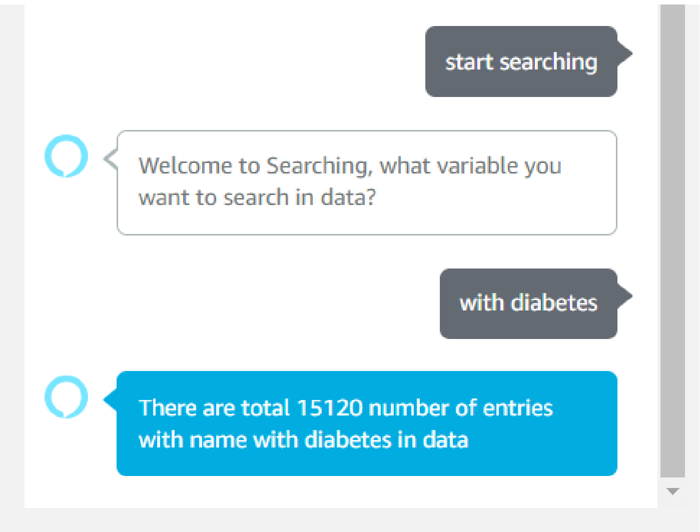
show geo

['GEO', 'Territories', 'Yukon', 'Northwest Territories including Nunavut', 'Northwest Territories', 'Nunavut']

Figure 4.9: Display (b)

4.7 Searching

The searching function is like a help to user to perform any searching in the dataset and also give its occurrences, as shown in figure 4.10. Alexa will prompt for the variable after starting searching and user have to provide with the search variable.



start searching

Welcome to Searching, what variable you want to search in data?

with diabetes

There are total 15120 number of entries with name with diabetes in data

Figure 4.10: Searching

Chapter 5

Conclusion & Future Work

The Database search approach using voice service was presented in this report. We made use of Amazon Web Services and Alexa skill kit to get things done. The results were conducted using different CSV format data files. This work has produced optimistic results. However, this research paper has scope of improvements which include:

- Creating a real time database system which can modify data in data file and create tables, charts, and perform other operations using data.
- Creating a stand-alone display interface to show the results in a more readable manner.

There is some scope of future work which can be done and it can act as great tool for predictive analysis and creating information from raw data.

References

[1] URL: https://www.prweb.com/releases/xpatterns/pubmed/prweb9726932.htm.

[2] URL: https://aibusiness.com/2019-trends-natural-language-processing/.

[3] URL: https://www.techgenyz.com/2018/09/20/latest-artificial-intelligence-advancements/.

[4] URL: https://towardsdatascience.com/natural-language-processing-in-apache-spark-using-nltk-part-1-2-58c68824f660.

[5] URL: https://searchbusinessanalytics.techtarget.com/definition/natural-language-processing-NLP.

[6] URL: https://www.forbes.com/sites/forbestechcouncil/2018/04/20/seven-artificial-intelligence-advances-expected-this-year/#7c3ea5f02652.

[7] URL: https://medium.com/@elvismiranda213/using-python-and-docker-for-data-science-c104e7b50497.

[8] URL: https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#3c0c313417b1.

[9] Foster Provost and Tom Fawcett. "Data Science and its Relationship to Big Data and Data-Driven Decision Making". In: 1 (1 Feb. 2013). DOI: http://doi.org/10.1089/big.2013.1508. URL: https://www.liebertpub.com/doi/10.1089/big.2013.1508.

[10] Zoubin Ghahramani. "Probabilistic machine learning and artificial intelligence". In: 521 (May 2015). DOI: http://doi.org/10.1038/nature14541. URL: https://www.nature.com/articles/nature14541.

[11] URL: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[12] Rajkumar Buyya et al. “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”. In: *Future Generation Computer Systems* 25.6 (2009), pp. 599–616. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2008.12.001. URL: http://www.sciencedirect.com/science/article/pii/S0167739X08001957.

[13] Prasetya Utama and Nathaniel Weir. “Voice-based Data Exploration : Chatting with your Database”. In: 2017.

[14] URL: https://en.wikipedia.org/wiki/Amazon_Alexa.

[15] URL: https://www.netcentric.biz/insights/2017/06/voice-control-your-jenkins/_jcr_content/responsivecontainergrid/fixed_responsivegrid/image_1463229232.coreimg.png/1540978516076.png.

[16] URL: https://developer.amazon.com/docs/custom-skills/create-the-interaction-model-for-your-skill.html.

[17] URL: https://image.slidesharecdn.com/yourfirstamazonalexaskill-161010202750/95/your-first-amazon-alexa-skill-11-638.jpg?cb=1476131430.

[18] URL: https://aws.amazon.com/.

[19] URL: https://d1.awsstatic.com/aws-answers/answers-images/serverless-image-handler-architecture.b09c96df4b6ae15054884c5c5ee0d869412de
png.

[20] URL: https://aws.amazon.com/lambda/features/.

[21] URL: https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121.

[22] URL: https://m.media-amazon.com/images/G/01/DeveloperBlogs/AmazonDeveloperBlogs/legacy/cloudformation_image12._CB520203781_.png.

[23] URL: https://developer.amazon.com/blogs/alexa/post/4e6db03f-6048-4b62-ba4b-6544da9ac440/the-scalable-neural-architecture-behind-alexa-s-ability-to-arbitrate-skills.

[24] URL: https://globalinforesearch.org/wp-content/uploads/2018/09/Amazon-Schedules-To-Deliver-8-New-Alexa-Powered-Devices.png.

[25] URL: https://developer.amazon.com/alexa-skills-kit.

[26] URL: https://aws.amazon.com/s3/.

[27] URL: https://cloudonaut.io/images/2016/06/fig_s3_overview.png.

[28] URL: https://aws.amazon.com/products/.

[29] URL: http://sublimedatasys.com/wp-content/uploads/2018/06/aws.png.

[30] URL: https://aws.amazon.com/lambda/features/.

[31] URL: https://www.itcentralstation.com/products/comparisons/apache-spark_vs_apache-storm_vs_aws-lambda.

[32] URL: https://developer.amazon.com/alexa.

[33] URL: https://m.media-amazon.com/images/G/01/mobile-apps/dex/alexa/alexa-skills-kit/asr/asr-lp_block-1._CB499320415_.png.

[34] URL: https://www.tomsguide.com/us/pictures-story/806-best-alexa-skills.html.

[35] URL: https://developer.amazon.com/docs/devconsole/about-the-developer-console.html.

[36] URL: https://3.bp.blogspot.com/-vXDssj0gRew/WDSoZ7cN_-I/AAAAAAAABG-k/mn8ZEdJfWogc_n5dDzwaSsQpY0ezsUIpwCK4B/s640/2016-11-22_12-18-32.png.

[37] URL: https://developer.amazon.com/docs/smapi/interaction-model-schema.html.

[38] URL: https://developer.amazon.com/docs/custom-skills/implement-the-built-in-intents.html.

[39] URL: https://developer.amazon.com/docs/custom-skills/best-practices-for-sample-utterances-and-custom-slot-type-values.html.

[40] URL: https://developer.amazon.com/docs/custom-skills/slot-type-reference.html.

[41] URL: https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html.

[42] URL: https://developer.amazon.com/docs/custom-skills/functional-testing-for-a-custom-skill.html.

[43] URL: https://developer.amazon.com/blogs/alexa/post/57d0bb9c-19a6-4c51-bfa2-fc6753d14b68/4-principles-of-conversational-voice-design.

[44] URL: https://developer.amazon.com/blogs/alexa/post/57d0bb9c-19a6-4c51-bfa2-fc6753d14b68/4-principles-of-conversational-voice-design.

[45] URL: https://developer.amazon.com/docs/custom-skills/dialog-interface-reference.html#delegate.

[46] URL: https://m.media-amazon.com/images/G/01/DeveloperBlogs/AlexaBlogs/default/image_3_cami._CB471576681_.png.

[47] URL: https://image.slidesharecdn.com/recap-creatingvoiceexperienceswithalexa
hk-revised-170214135009/95/creating-iot-solutions-with-serverless-architecture-alexa-8-638.jpg?cb=1487080388.

[48] URL: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html.