Adheetee: A Comprehensive Bangla Virtual Assistant

Conference Paper · May 2019

DOI: 10.1109/ICASERT.2019.8934903

5 authors, including:

Syed Mohidul Islam
United International University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Most Fowziya Akther Houya
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Syed Mynul Islam
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Salekul Islam
United International University
98 PUBLICATIONS **746** CITATIONS
SEE PROFILE

Adheetee: A Comprehensive Bangla Virtual Assistant

Syed Mohidul Islam, Most Fowziya Akther Houya, Syed Mynul Islam, Salekul Islam, Nahid Hossain
Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh
Email: isyedmohidul@gmail.com, fowziyaboshra@gmail.com, syed.mynul1@gmail.com, salekul@cse.uiu.ac.bd, nahid@cse.uiu.ac.bd

***Abstract*— A Virtual Assistant is an application program that can understand natural language (audio and text) commands and perform tasks for the user. The importance and usage of the virtual assistant are growing exponentially worldwide. In this paper, we have demonstrated a comprehensive design and development methodology of a Bangla virtual AI assistant 'Adheetee (Erudite)' for smart devices such as smartphones and personal computers. Bangla is the native language of Bangladesh and some states of India with millions of native speakers worldwide. There are some popular virtual assistants available in English and other languages but not in Bangla. Therefore, using a virtual assistant become very difficult for Bangla speaking people around the world. Moreover, a large portion of Bengali speaking people is illiterate. This makes using a virtual assistant more difficult. This gives us the motivation to build a comprehensive virtual assistant for Bangla language. We have covered vast varieties of most frequent commands in our system with the accuracy of 94.065%.**

Keywords—Adheetee, Virtual Assistant, Bangla, Commands, API

I. INTRODUCTION

A Virtual assistant makes tasks of a person in a smartphone or a personal computer easier by just giving a voice command or a written command. The usage of a virtual assistant is presumed to hit 1.8 billion worldwide in 2021 [1][2]. People can get benefits of 21st century's technology by using a virtual assistant as for using it no extra knowledge is required. A user can use it just like they communicate with a human being. Intelligent natural language processing makes the system understand the natural language commands of a human being [3]. A virtual assistant can be used to set a reminder, sending email, know weather information, search information in Google, update to-do list etc. Bangla is the 6th most widely spoken language with approximately 250 million total speakers worldwide [6]. However, there is no virtual assistant yet developed for Bangla language. Euromonitor International reports only 18% of the total population in Bangladesh can understand and speak English [7]. In addition to that 27.2% of the total population is illiterate who cannot even read or write Bangla language [8]. Due to the low literacy rate and ineptness of English knowledge among the Bangla speaking people we are not being benefited properly by the English virtual assistants. Therefore, the authors decided to propose a comprehensive Bangla virtual assistant which takes both verbal and text Bangla input as a command. According to Newzoo's Global Mobile Market Report 2018 [11], there are around 27 million Smartphone users in Bangladesh. Our proposed system can help these enormous users to be able to utilize features of their smartphones more easier and faster way in their mother tongue.

Previously, a considerable amount of work had been done in virtual assistant in English and other western languages such as Google's Google Assistant, Apple's Siri, Microsoft's Cortana, and Samsung's Bixby etc. Although most of them have the same functionalities, each of them has some new features and abilities [5]. However, no virtual assistant has been designed or developed in Bangla language yet. Note that there are some chatbots available in Bangla. T.D. Orin proposed Bangla chatbot named Golpo in 2017, which can have basic text conversations [10]. In 2018, Anirudha Paul et al. proposed a closed domain contextual chatbot framework for resource-poor languages including Bangla [9]. However, these are only text chatbots for general conversations and thus have very limited functionalities.

In this paper, we have proposed a comprehensive Bangla virtual assistant for smart devices. We have named our system 'Adheetee' which is a Bangla word, meaning 'Erudite (having or showing great knowledge or learning.)'. It contains a large set of most frequent commands and variations of those commands. Our system is consists of a large database containing several types of information about different commands and their potential responses. We have distributed our commands as basic commands and core commands. Basic commands handle a basic conversation like a chatbot. However, our basic conversations are not just introductory communications; we have stored a user's important information through the conversation for future references, make the system more customized and to provide responses based on context. In the core commands, the system handles more complex device oriented activity. Handling varieties of the same command is a difficult task, however, we have used Named Entity Recognition (NER), Keyword Extraction (KE) and Cosine Similarity (CSM) algorithms to identify and respond accordingly for command variations. We have built our system to be able to respond by itself using its intelligence and the information stored in its database. However, to handle unknown commands and fetch the latest information the system sometimes depends on external API calls.

The paper is organized as follows: In section II, we describe the proposed system and step by step demonstration of the system diagram with proper examples and algorithm. The paper demonstrates experimental results in section III with examples. Section IV interprets performance from the experiments, while section V concludes the paper with limitations of our system and future work.

II. PROPOSED METHOD

Our system takes both voice command and text command as input. A person can wake up our system just by calling "হে অধীতী (Hey Adheetee)" or "ওহে অধীতী (Hello Adheetee)" or simply "অধীতী (Adheetee)". On the other hand, a user can put written text as a command. For voice command, we translate the speech into text by using a Speech to Text (STT) system. We have used Google's STT

API to do this work [12]. Then, we extract keywords from the text command. We need to extract the keywords from command due to the fact that the commands are given in natural language. Therefore, a different user can speak or write the same command in different ways. Moreover, even the same user can speak or write differently in different time periods. For example, if we observe the following commands:

a. *আমাকে আবহাওয়ার খবর বলুন?* (Give me weather update.)

b. *আজকের আবহাওয়া কেমন?* (How is the weather today?)

c. *বাহিরের তাপমাত্রা কত?* (What is the temperature outside?)

d. *অধীতী, আজ বাহিরের তাপমাত্রা কত?* (What is the temperature outside today?)

e. *কাল আমাদের এলাকার তাপমাত্রা কেমন থাকবে?* (What will be the temperature tomorrow in our area?)

We can see that all of these different commands actually mean the same thing and it is requesting for the weather update. Thus, here, our keyword is আবহাওয়া (weather) for the examples a and b, and তাপমাত্রা (temperature) for the examples c, d and e. Based on these keywords, we determine the type of commands such as whether it is a basic command or a core command. We have divided commands into two different phases. These are,

1. Basic Commands
2. Core Commands

In basic commands, we handle commands with basic conversations. Such as asking for a name and storing the name. Having basic knowledge question answering conversation goes to basic commands. In this case, we generate a cosine similarity measure [14][15] between the keywords and commands stored in our knowledge base to determine the command category. This helps us to determine commands category of the same command in different variations. On the other hand, more complex commands such as a command that need actions goes to core commands such as opening an app, setting up an alarm, searching specific information on the Internet etc. Figure 1 demonstrates the system diagram of our proposed system.

On receiving a command, the system determines whether it can respond by itself or we need to call an external API to fetch the information from other sources. In some commands, we do not depend on external APIs such as, all basic commands and commands that the system can handle itself. In this case, our system collects data from our knowledge base and/or store data from the user for future references. On the other hand, in some commands, we need to call an external API to do our work. We also determine based on keywords whether we need to translate the keyword into English since some external APIs do not support Bangla words as input parameters. In such cases, we apply machine translation API to translate the Bangla text to equivalent English text and pass the English text as a parameter of the external API. For machine translation, we have used the Google Translate API [13]. In some cases, Google Translate API provides some garbage data such as '#?' with the result. Thus, we have developed a program based on several regular expressions to filter out these unwanted garbage data.

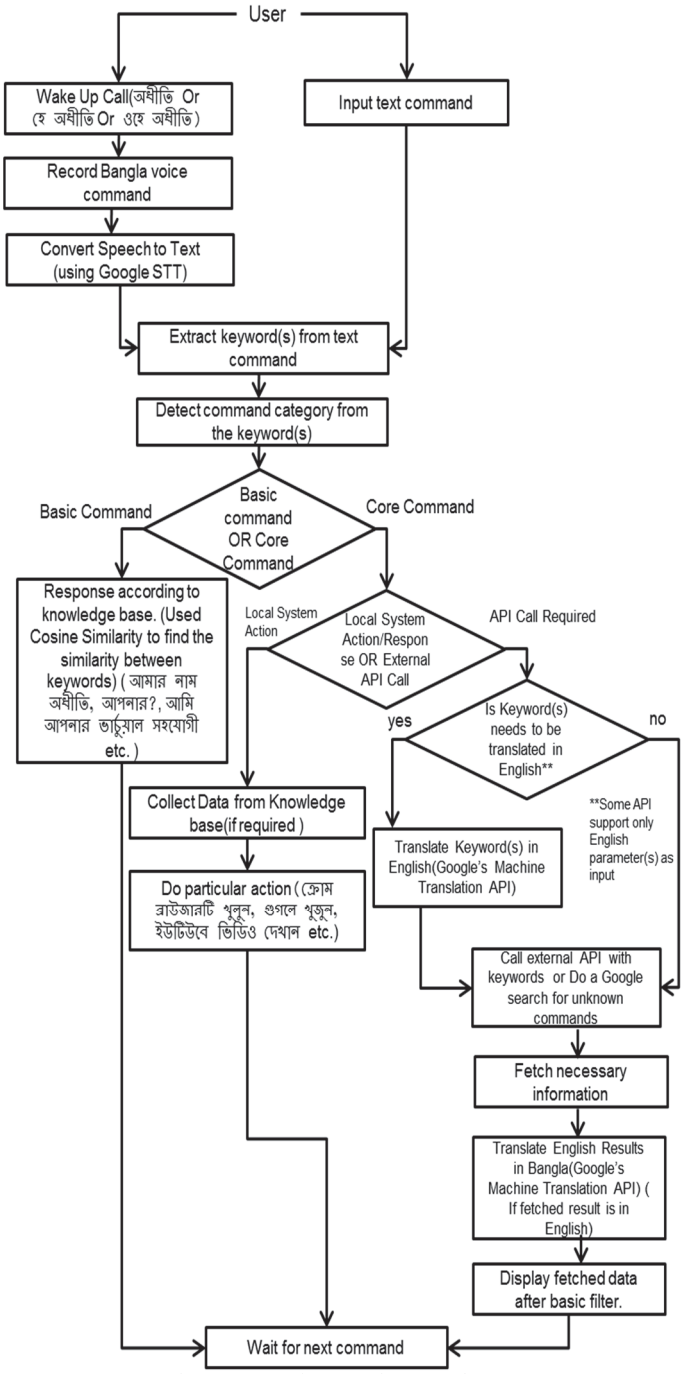

Fig.1 System Diagram of Proposed System

The algorithm of our proposed system is given below.

Algorithm 1: Algorithm of proposed system

```
IF wakeup call OR press the voice button THEN
    RV ← record voice command
    IN_TXT ← convert speech command to text(using Google's STT)
ELSE
    IN_TXT ← read text command
END IF
KEY ← extract keyword(s) by analyzing IN_TXT
CAT ← detect command category using KEY
IF CAT is a basic command THEN
    CS ← measure cosine similarity between CAT and keywords from knowledge base.
    based on CS determine closest command and give response accordingly.
```

```
KB ← Store basic user information in knowledge base for future reference (such as user's name, birthdate, favorite color)
ELSE
    LORAPI← detect whether the command can be handle by our system or we need external API call
    IF LORAPI indicate our system call handle THEN
        collect data from knowledge if required
        do particular action based on command
    ELSE
        ENG← detect whether translation of keyword(s) to English required or not
        IF ENG indicate translation required THEN
            translate keyword(s) to English (using Google's Machine Translation)
        END IF
        call external API with the keyword(s) as parameter
        JSON←fetch necessary information and display accordingly
        IF JSON returns English texts THEN
            convert JSON to Bangla (using Google's Machine Translation)
        END IF
        Extract necessary information from JSON
        Display JSON
    END IF
END IF
```

We have created a dataset of keywords and their corresponding actions. However, if someone put command with a keyword outside of our dataset of keywords, we search the whole command in Google and show the result accordingly. This dataset also let us know whether we need to call an external API to fetch information or our knowledge base is sufficient to respond to the command. In our knowledge base, we have developed an SQLite database and we have several JSON and text files which contain different types of data from different domains and functionalities. The reason behind this distributed knowledge base is to ease of access and to make modification easier. We have an index JSON file that contains commands, keywords of commands, flag that indicates whether it is a basic command or core command, flag that indicates whether we candle this by our system or we need an external API call, flag that indicates whether we need to translate the command keyword(s) in English or not and some other necessary information. This file makes our work easier to decide what to do next. On each external API call, we get JSON formatted data as a result and we keep a local log of the command and fetched data to enrich our system. However, we also store the latest data whenever we find new data for the same command to keep the system up to date.

III. Experimental Results and Analysis

In this section, we have demonstrated the output of different commands in our proposed system and analysis of the results. We have distributed the commands into Basic and Core commands. In total, we have processed 221 commands.

a. Basic Commands

In basic commands, we have processed commands that people use for an initial conversation, to know basic information about our system and some basic mathematics. We have designed the responses as natural as human conversation and our system stores important details for contextual responses. At the very beginning of a conversation with our system, if a user says introductory words such as 'Hi', 'Hello' or 'Hello Adheetee', our system responds mentioning the user's name. Knowing the name of a user is very important for natural friendly conversation. Thus, we address the user as '(বন্ধু) friend' until we know the name of the user. After storing the user's name, we always call the user with his/her name. We have processed 101 basic commands and variations of these commands so that it responds in a natural way just like a human does. Examples of basic commands are:

a. আপনার নাম কি? (What is your name?)

b. আপনাকে কে সৃষ্টি করেছে? (Who created you?)

c. আপনি ছেলে না মেয়ে? (Are you a boy or a girl?)

d. আমার নাম করিম, আপনার নাম কি? (My name is Karim, What is your name?)

e. অধীতী, আমার প্রিয় রঙ লাল এটা মনে রাখুন? (Adheetee, Remember my favourite colour is Blue.)

f. অধীতী, আমার নাম কি বলুনতো? (Adheetee, Tell me my name?)

g. আপনার প্রিয় রঙ কি? (What is your favourite colour?)

h. ২৫ কে ৩ দিয়ে ভাগ করলে কত হয়? (What is 25 divides by 3?)

Figure 2 shows a very initial conversation with a user.

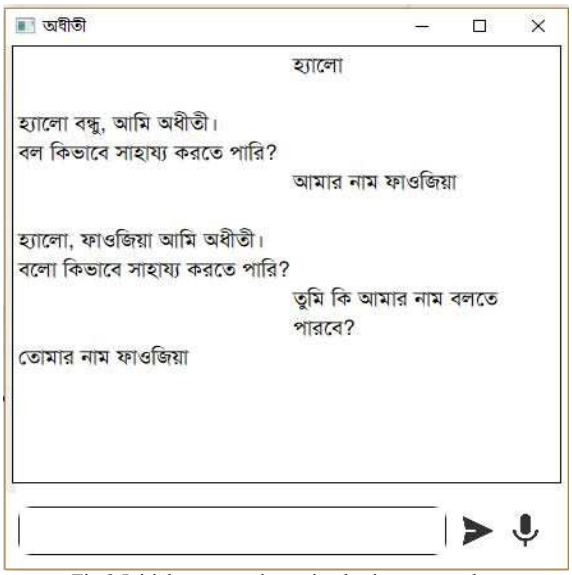

Fig.2 Initial conversation using basic commands

b. Core Commands

In core commands, we have processed commands that need particular actions such as opening a browser, setting an alarm, opening a file, playing a song etc. Core commands are mostly action based with very little conversation. We have processed 120 core commands in our system. Examples of core commands are,

a. অধীতী, দুপুর দুইটার জন্য অ্যালার্ম সেট করুন। (Adheetee, Set an alarm for 2 PM.)

b. অধীতী, ক্রোম ব্রাউজারটি খুলুন। (Adheetee, Open the Chrome browser.)

c. আমার প্রিয় গানটি বাজান। (Play my favourite song.)

d. ইউটিউবে ইংরেজি শেখার ভিডিও চালান। (Play video tutorial of English in Youtube.)
e. মোহাম্মদপুর থেকে মতিঝিল যাব কিভাবে? (How to visit Motijheel from Mohammadpur?)
f. অধীতী, রাশিয়ার রাজধানীর নাম কি? (Adheetee, what is the capital of Russia?)
g. আজ আমার এলাকার তাপমাত্রা কত? (What is the temperature today in my area?)
h. করিম কে ইমেল কর যে আমি আজ অফিসে যাব না। (Send an email to Karim saying I will not go office today.)

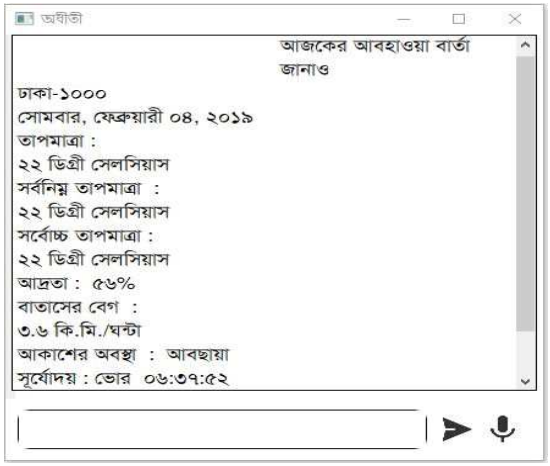

Fig.3 Example of weather command

Figure 3 shows how our system responds for a weather command. We fetch information from internet based on GPS (Global Positioning System) location. As weather information, we fetch different information including temperature, sunset, sunrise, and humidity. Our system shows all this information for a generic weather command. However, if a user asks for specific information the system shows only that specific information.

Sending an email is one of the most important tasks in our daily life. Thus, our system makes it easier to send an email using simple Bangla command. A user needs to say the intended email address or just the name of a recipient with the text.

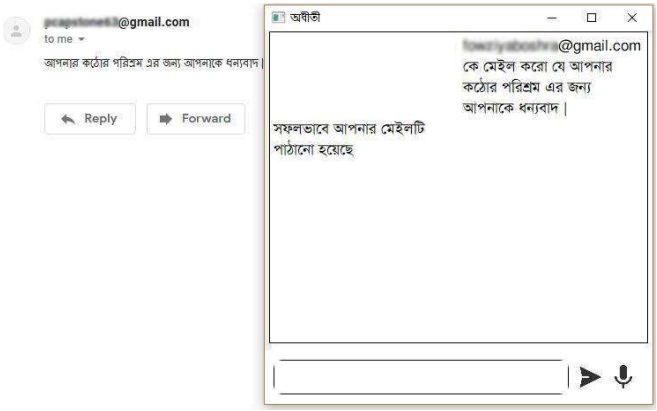

Fig.4 Example of sending email command

In order to send an email by just a name of a recipient, a user must have to give a command anytime earlier so that our system store the email address corresponds to that name. However, if our system does not have an email address for a name it automatically asks for the email address later. Figure 4 shows the response to an email command.

Travelling in a large city such as Dhaka is difficult. In order to go to a new place we often depend on the map. Our system makes it easier to get a map direction from one place to another in the Bangla language. We have used Google Map to display the direction from a source to a destination. For a simple Bangla command such as 'আমাকে মিরপুর ১২ থেকে উত্তরা যাওয়ার রাস্তা দেখাও। (Show me the direction from Mirpur 12 to Uttara.)'. In this case, we have used an NER (Named Entity Recognition) algorithm [4] and gets the name of source and destination. After that, we call the Google Map API using these names in appropriate parameters and finally display the route from the source to the destination. If the user says a command with no source but only destination such as 'আমি বাড্ডা যাব কিভাবে? (How to go Badda?)', we find the source location using the GPS. Figure 5 shows a direction command.

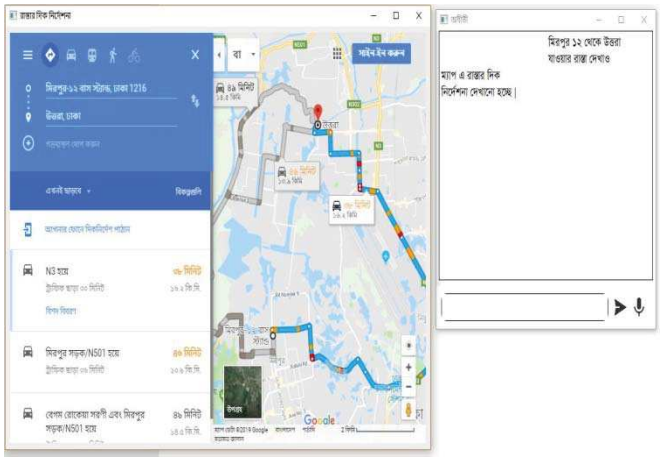
Fig.5 Example of seeking direction in map command

To open a browser, a user needs to mention the name of the browser. Our system opens the browser immediately if the browser is available in the device. Figure 6 shows a command 'অপেরা ব্রাউজারটি ওপেন কর। (Open the Opera browser.)' that open the Opera browser.

Fig.6 Example of opening a browser command

If a user searches for a video or wants to play a song he can command with appropriate name and source such as Youtube, Amazon Prime, Vimeo etc. If a user asks for a video without mentioning the source of the video, then by default we search the video on Youtube. In Figure 7, we

have shown an example of command seeking for a video on Rubik's cube without mentioning the source.

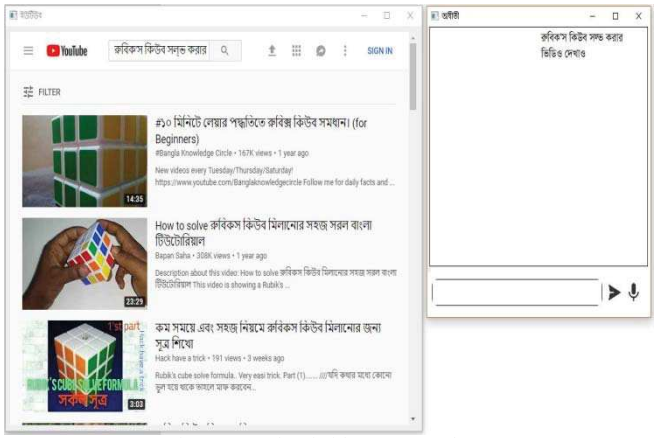

Fig.7 Example of video command

Our system can fetch any general information from different sources. It can give answers to almost any general knowledge questions. We have an enormous database that contains all the basic general information of all the countries in the word. This contains basic information such as counties name, capital, the biggest city, size, celebrities, poets, the name of the presidents, national sports etc. If a user asks for such basic commands, our system first tries to give answers from its database. However, if there is no data available for specific countries our system fetches the information from the internet. Figure 8 shows some basic general knowledge commands.

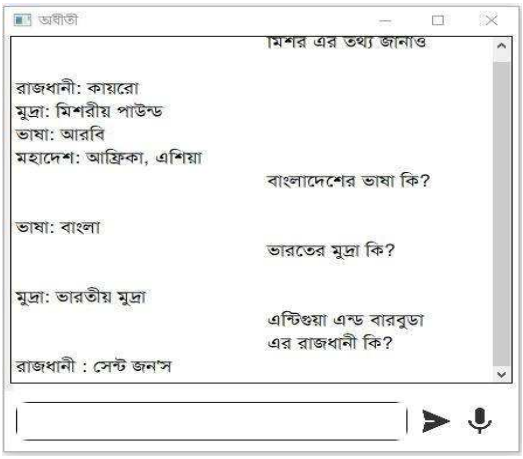

Fig.8 Example of responses to general knowledge commands

A user can search a term in Google using our system with Bangla command. Moreover, if a user asks something that cannot be answered by our system itself, then our system search the command or term in Google directly and show the search result of Google. Figure 9 shows a search result of Google by our system.

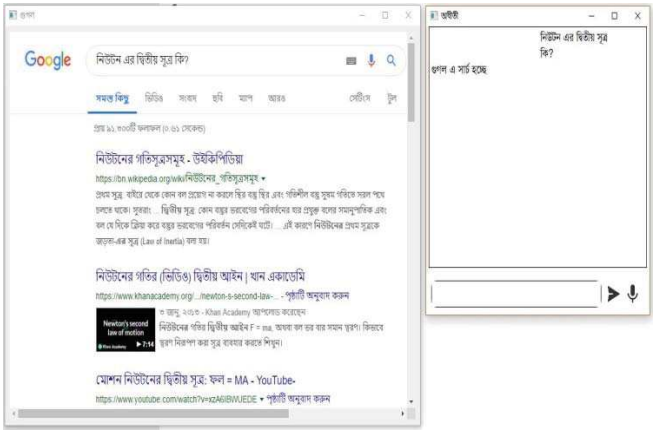

Fig.9 Example of Google search for unknown term by our system

IV. Performance Analysis

In order to calculate the performance of our system, we have tested our system in two phases. In the first phase, we have tested our system with some basic commands known and unknown to the system such as, "আপনার নাম কি? (What is your name?)" "আপনি কি কি করতে পারেন? (What can you do?)" etc. In this phase, we have determined our system's ability to have a basic conversation with a user. In the later phase, we have applied our core commands such as "অধীতী, ক্রোম ব্রাউজারটি খুলুন (Adheetee, open the Chrome browser.)", "অধীতী, ধানমন্ডি থেকে বাড্ডা যাওয়ার ম্যাপ দেখান (Adheetee, show the direction in map from Dhanmondi to Badda)" etc. to test our system's ability to do complex things with a device. In this case, we have provided all possible variations of a command a user can ask to our system. We have considered a test success if our system correctly identified our command and correctly provided the expected output. On the other hand, if our system failed to detect our command correctly or it provided incorrect or irrelevant output then we considered the test as a failure. Table I demonstrated the test summary of our system.

TABLE I. Performance Summary of Our System

Phase	1	2
Name of Phase	Basic Commands	Core Commands
Total Commands Tested (including all variations)	102	112
No. of Correctly Identified and Responded Commands	99	102
No. of Incorrectly Identified or Responded Commands	3	10
Error Rate	2.94%	8.93%
Accuracy Rate	97.06%	91.07%

TABLE II. Average Error and Accuracy Rate

Average Error Rate	$\frac{2.94\% + 8.93\%}{2}$	= 5.935%
Average Error Rate	$\frac{97.06\% + 91.07\%}{2}$	= 94.065%

In Table I, we have demonstrated the test results of both phases and Table II demonstrates the average error rate and accuracy rate of the system. The average accuracy rate of the system is 94.065% with the error rate of 5.935%. Since our proposed virtual assistant is the first Bangla virtual assistant

and there is no Bangla virtual assistant available to compare with, we have compared our system with existing English virtual assistant. Figure 10 shows a comparison of the accuracy rate between English virtual assistants and our system 'Adheetee'. The accuracy rate of these English virtual assistants was reported in an article of Business Insider [16].

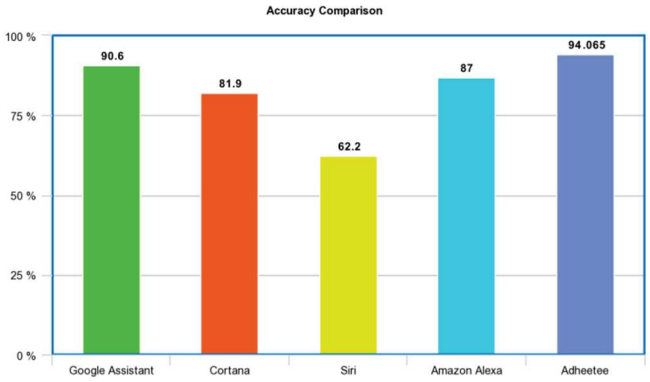

Fig.10 Accuracy comparison between English virtual assistants and Adheetee.

V. Conclusion and Future Work

This paper demonstrates a comprehensive Bangla virtual assistant. We have demonstrated each and every step of building the system. Moreover, we have shown different experimental results and analysis of the results and finally, we have demonstrated the test results. We find our system very impressive according to our test results.

Although we have reached our goal, the system has some limitations. Such as, our system mostly depends on external APIs to fetch information which makes the system dependent on other systems. Since the system depends on the internet to fetch most of the information internet speed may affect the performance of the system. Finally, we have covered a limited number of commands at this stage which makes the system less versatile.

In future, we are planning to do a user experience survey according to HCI discipline based on a robust online and offline questionnaire. We are developing some APIs to reduce the dependency on external APIs. Moreover, we are developing a mobile app for both Android and IOS platforms. Handling more frequent commands is our most important future task. We are planning to test the system with more complex commands and hence increase the accuracy rate of the system.

References

[1] Hoy, Matthew B. (2018). "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants". Medical Reference Services Quarterly. 37 (1): 81–88. doi:10.1080/02763869.2018.1404391

[2] Rudolph, Stacey. The Growth in Usage of Virtual Digital Assistants [Infographic]. Retrieved from https://www.business2community.com/infographics/the-growth-in-usage-of-virtual-digital-assistants-infographic-02056086

[3] A. Bodart and C. Condon, "Virtual assistant," Feb. 22 2007, uS Patent App.11/206,829.

[4] Chowdhury, Shammur & Alam, Firoj & Khan, Naira. (2018). Towards Bangla Named Entity Recognition. 10.1109/ICCITECHN.2018.8631931.

[5] Varma, S. (2018, June 14). Siri vs Google Assistant vs Alexa: Which Is the Smartest Virtual Assistant in 2018? Retrieved from https://gadgets.ndtv.com/apps/features/siri-vs-google-assistant-vs-alexa-which-is-the-smartest-virtual-assistant-in-2018-1867292

[6] Hays, Jeffrey. BENGALIS. Retrieved from http://factsanddetails.com/india/Minorities_Castes_and_Regions_in_India/sub7_4b/entry-4198.html

[7] Pinon, Robert; Haydon, Jon (December 2010), The Benefits of the English Language for Individuals and Societies: Quantitative Indicators from Cameroon, Nigeria, Rwanda, Bangladesh and Pakistan, Euromonitor International Ltd

[8] "Bangladesh - Literacy rate 2017." [online] countryeconomy.com. Available at: https://countryeconomy.com/demography/literacy-rate/bangladesh

[9] Paul, Anirudha & Latif, Asiful & Amin Adnan, Foysal & Rahman, Mohammad. (2018). Focused domain contextual AI chatbot framework for resource poor languages. Journal of Information and Telecommunication. 1-22. 10.1080/24751839.2018.1558378.

[10] Orin , T. D. (2017). Implementation of a Bangla Chatbot (Unpublished Bachelor Thesis). BRAC University, Dhaka, Bangladesh.

[11] "Top Countries/Markets by Smartphone Penetration & Users. 2018." [online] newzoo.com. Available at: https://newzoo.com/insights/rankings/top-50-countries-by-smartphone-penetration-and-users/

[12] "Cloud Speech-to-Text - Speech Recognition | Cloud Speech-to-Text API | Google Cloud." Google, Google, cloud.google.com/speech-to-text/.

[13] "Cloud Translation API Documentation | Cloud Translation API | Google Cloud." Google, Google, cloud.google.com/translate/docs/.

[14] A. Dhar, N. Dash and K. Roy, "Classification of text documents through distance measurement: An experiment with multi-domain Bangla text documents," *2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall)*, Dehradun, 2017, pp. 1-6. doi: 10.1109/ICACCAF.2017.8344721

[15] The Cosine Similarity algorithm. [online] www.neo4j.com. Available at: https://neo4j.com/docs/graph-algorithms/current/algorithms/similarity -cosine/

[16] Dunn, Jeff. "It Looks like Apple Has Some Work to Do If It Wants Siri to Be as Smart as Google Assistant." Business Insider, Business Insider, 2 June 2017, www.businessinsider.com/siri-vs-google-assistant-cortana-alexa-knowledge-study-chart-2017-6.