Inception of a containerized multi-purpose Chatbot

Sanjitha Gowda[1]*, Preetham TP[1], Dr. Mamatha G S[2], Prof. Anisha B S[2]

[1,2] Department of Information Science and Engineering, RV College of Engineering®, Bengaluru

---***---

Abstract - *Interaction of humans with computers and other digital media has been increasing rapidly every day. One way to make this experience better for the users is to make them their interaction with computers as similar as possible to their interaction with humans. Chatbots are used to serve that purpose. They are major milestones that help humans and the world move closer to automation. Microservices are a method of software development in which services are built to perform a single function or a small set of closely-related functions i.e, services are built to be extremely modular. Containerization of a microservice is the packaging of the entirety of the runtime of the microservice into a lightweight, platform-agnostic image which can then be deployed across a multitude of hardware platforms. Although the ubiquity of container technology has exponentially increased in the recent past, it is not a new concept. This paper deals with the designing of a chatbot using the various techniques that come under natural language processing using Python followed by the containerization of the chatbot to make it as portable and consistent as possible also so that new modules can be added easily.*

Key Words: Stopwords, ML, NLP, chatbot, Microservice, serverless.

1. INTRODUCTION

Chatbots are computer programs that converse with users using natural language. This conversation can be primarily of two mediums, that is textual and acoustic. Chatbots are becoming familiar owing to an increased use in practical applications such as information acquisition, customer interaction and dialogue-based systems. Natural language processing is a paradigm of software technology which allows bots to comprehend natural language and perceive users' requirements gracefully.

Small businesses have a high ratio of number of customers to employees. Thus, not every customer can be provided individual attention. Such businesses require the power of automation in their customer support systems. Through automation, the chatbot can play the role of an employee attending to a customer personally. Chatbots are very effective as they lessen resources required, can function with meagre human intervention and also supply huge profit to the business. Chatbots are a growing trend in the field of business, due to its easy development and setup, cost-efficiency, high levels of productivity than their human counterparts, marketing strategies.

The software development methodology used to build chatbots supporting several, different domain queries and actions at one single interface to the user is achievable through microservice architecture. Microservice architecture mainly deals with breaking down of a system into several services which are modular in nature, and also providing interface between such services. It is essentially a loosely coupled system comprising of multiple services handling separate functionalities independently. Microservice architecture allows for easy scalability, maintenance and extensibility. It can be hosted using a serverless paradigm which is a pay-per-use facility and it isolates maintenance from the user. Through this software development methodology, various independently operating services can be combined loosely to come together as a single system with multiple capabilities.

Fig 1. showed below shows a simple representation of microservice architecture. In our case, each microservice represents a chatbot that is built for a particular application

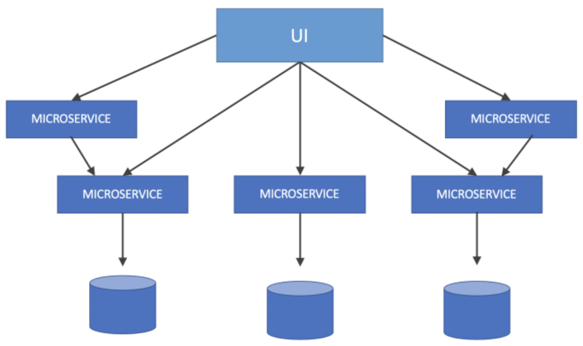

Fig -1: Microservice Architecture

2. MOTIVATION

The existing human interaction systems are less-efficient, require high maintenance and require excessive manual effort. Thus, the need for a chatbot is vital. This paper proposes a methodology to build such an intelligent chatbot following the microservice architecture of software development, thus alleviating the costs to the company and contributing largely to the profit of it.

3. LITERATURE SURVEY

[1] discusses about a hybrid learning method which combines imitation and reinforcement learning to perform as a better agent in task-oriented dialogue systems.

Initially, they train the system using a supervised learning mechanism on a dialogue corpora, then have implemented reinforcement learning to occur based on user feedback giving positive rewards for success and negative rewards for mistakes, thus leading to a better agent performance faring above each individually implemented model.
[2] discusses about building a dataset which is a new fully-labeled collection of human-human written conversations which span over multiple domains and are of magnitudes larger than previously available dialogue corpora. It serves as a great benchmark for a range of dialogue tasks such as dialogue-state tracking, dialogue-context-to-text generation.
[3] states that majority of conversations the bot sees in its lifetime is only after it is trained with a corpora and deployed into use, leaving a vast amount of conversation training data untapped. They discuss about a self-feeding mechanism wherein the bot generates new training samples from the conversations it has with the users to improve its accuracy. Using this mechanism, the dialogue agent imitates human responses when satisfied to act as a feedback for the chatbot on newly generated training data set.
[4] Provides a detailed overview about the performance of a virtual machine and the way docker and containerization can be used to improve the performance of the system. It contrasts the working of virtual machine with that of docker.
Web applications and the rapid growth in the area of web development has greatly benefitted people's work and life. With the advent of cloud computing resources can be provisioned on-demand. In [5], the author discusses about a platform which can use docker container technology to auto scale web applications on hybrid cloud, by adding or deleting docker containers, scheduling containers etc.
[6] Provides an overview about how the container ecosystem works and discusses the security implications of using docker with the help of realistic use cases. It points out several vulnerabilities affecting the usage of Docker, and discusses further research.

4. PROBLEM STATEMENT

With increasing advent in computers and digital technology, the percentage of the population that interacts with the computers also has increased. This shows a pressing need for a system that can address the user queries faster and in an efficient and user friendly manner. Chatbots help to serve this purpose.

One single chatbot for various applications isn't very effective as it will need to access different DBs which will not only have security implications but will also affect the performance of the chatbot. This calls for the development of chatbots in a modularized fashion using microservices.

The deployment of chatbots on various environments also is a tedious process as it will need a lot of dependencies. This contributes to a lot of the unnecessary complexities software development has become synonymous with.

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

5. OBJECTIVE

This paper focuses on 3 major aspects.

a. Building a chatbot.
b. Making use of microservice architecture to support chatbots that can be used for various applications.
c. Containerization of chatbot.

6. APPLICATIONS

The various applications of a chatbot are:

a. *Product suggestions*

Many consumers may want to buy a particular product, but aren't aware of its name or identity, then they can describe the product to the bot, which understands the conversation and suggests products matching that description.

b. *Customer support*

Chatbots can also take the role of customer support centers by conversing about the problems faced by the user, then retrieve solutions or resources that match to the user's dilemma. These type of chatbots are rather soft and pleasant spoken so they do not infuriate the customer.

c. *Weather*

Chatbots satisfying this application are meant to retrieve weather data from online sources and also provide detailed analysis of the weather for past few days in form of a conversation.

d. *Personal Finance Assistance*

Chatbots can act as stock brokers, suggesting and convincing shareholders to sell or buy stocks, to make profit by using sophisticated data analysis algorithms. Chatbots can also be permitted to access bank accounts and display current balance or display bank statements too.

e. *Schedule a Meeting*

With so many meetings occurring, scheduling a new one can be a hassle. This can be taken care by a chatbot if given required authentication accesses to retrieve each meeting participant's schedules to elaborately predict the convenient time for all participants to attend the meeting.

7. METHODOOGY

There are mainly three phases proposed for the development of a containerized chatbot as elucidated below:

7.1 Chatbot building

Fig. 2 shows the major steps that are involved in the building of the chatbot. This is the first phase in the proposed methodology. It involves creation of a chatbot that can accept user queries, and then return results for that query

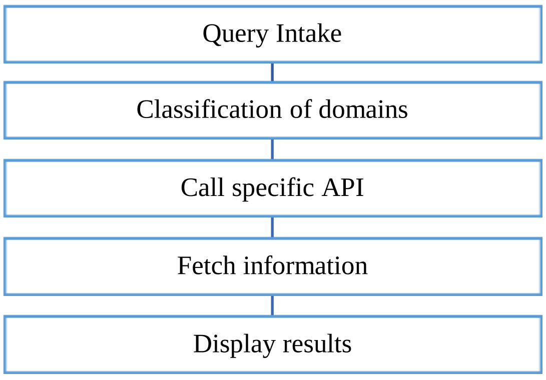

Fig -2: Building the chatbot

This phase can be divided as shown above in Figure 2 into five services. They are:

a. Query Intake – This operation deals with the task of obtaining query as input from the user. The user can enter the query at either the chatbot platform or the web platform interfaces. This service solely takes a question or text and transmits it to the next service for classification.
b. Domain classification – This service deals with the classification of a given question into a specific domain. This classification is carried out by using Machine Learning (ML) methodologies to predict domain of a query. Initially, the machine learning model is trained on different set of queries. So, in this step, the stop-words in the query are discarded and other various Natural Language Processing (NLP) operations like stemming, tokenizing, and named entity recognition is carried out, while the keywords which can act as features to identify a domain are retrieved. These features are sent to the ML model to predict the domain of the query accurately.
c. Specific API call – This service takes output from the classifier and based on the domain classified, it transmits the query and the relevant parameters to another service which is domain specific. At that service, a suitable API is called to get information.
d. Fetch information – This service deals with obtaining the required information from the API call and filtering the results as required by user.
e. Display results – The last service is to format the obtained raw information to an user-readable format and then sends the result back to the source where the question was queried by the user initially.

This marks the completion of building the chatbot. Now, to containerize it, the next two phases is to be carried out.

7.2 Microservice Architecture

In this phase, the chatbot is categorized into multiple services which are independent and modular in nature, facilitating the foundation for a microservice architecture. The interface service of the chatbot, query input service of the chatbot, domain classifier service, multiple domain specific API services, data retrieval service and output service can all be separated and made to work individually. Each service now individually takes a given input and yields an output which is to be standardized to make code maintenance and updating easier in the future. Each specific domain service can take care of its functionalities by exposing a port which listens to the API calls and synchronizes it carefully.
Each service that was individualized is now built into REST API's to avert the problem of storing state at all times. All the services are now listening at separate ports, where the REST API will be called for processing. With the use of separate databases and accesses, all the modular services can communicate with each other, giving rise to a serverless chatbot platform supporting multiple domains.
Thus, the microservice architecture is built for the chatbot. Now, containerization of microservices so that, dependencies and overhead can be mitigated, is taken care of in the next phase.

7.3 Containerization of Chatbot

This particular phase includes packaging the applications and their dependencies into containers.

The container technology used in this project is Docker. Docker is a PaaS offering, which provides the tools necessary for the development, running and shipping of applications in containers. Docker Engine on the other hand is a client-server application and it provides the functionalities of the Docker platform. It includes the Docker daemon, a REST API interface that directs the daemon and a CLI that can be used to execute commands. Docker daemon is the continual process that runs in the background and manages the containers that are running on the system. Docker CLI has the commands that are needed to build docker images, create new containers and run commands in the active and running containers. The platform also includes DockerHub, which is a container

image registry that provides the base images which can be used to run applications.
The following steps are to be followed:

a. Download an existing base image from the repository.
b. Add the required dependencies in the docker image file using a container build file.
c. Build the docker image
d. This will create a container
e. Run the container

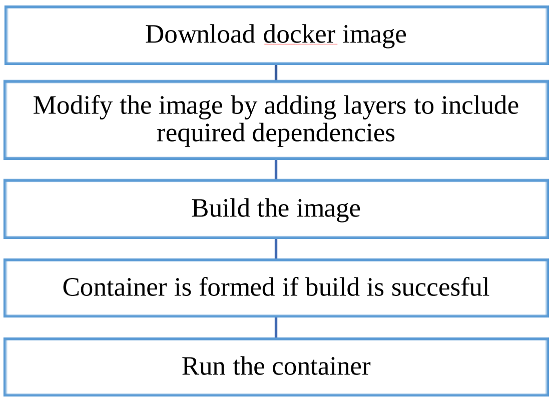
Download docker image
Modify the image by adding layers to include required dependencies
Build the image
Container is formed if build is succesful
Run the container

Fig -3: Containerization of Application

8. CHALLENGES

8.1 Natural Language Processing

Making the chatbot understand human language is a major challenge. Few basic queries and basic sentences can be handled by the chatbot. But as the query length increases it's hard for the chatbot to predict the responses. Tuning the chatbot to give correct responses to all queries is still a work in progress.

8.2 Machine learning

Another major challenge is to get our systems to obtain the correct response to the query that is posed to them. Various AI techniques can be used for this purpose. But we need to maintain the right balance of complexity and efficiency.

8.3 Support required for Docker

Docker is a relatively new and growing technology. The amount of community support that is available for docker is very less. Hence for a beginner to start working with docker, it might be a very difficult task

[12] provides more details about the challenges involved in construction of a chatbot whereas [6] provides details about the challenges of Docker.

9. CONCLUSION

The chatbot built here overcomes all the drawbacks faced by existing systems as the bot can support multiple domains with a single interface, efficient usage, cost-effective and makes huge profit to small businesses. This process takes less time and less resources by a factor of five times compared to the existing interface, so the motivation to use this chatbot is amplified. The chatbot interface is friendly to users as it makes use of NLP operations and can also obtain results faster thus promoting increased usage by the users. By using this chatbot for supporting a small business with multiple fields of interest can be rewarding.
This paper attempts to provide a concise summary of microservice architecture and chatbots. It briefly brings out the inherent challenges involved in such intelligent systems. It surveys other related works to identify the shortcomings and then propose solutions that overcome such limitations. Mainly, the paper proposes a methodology which can be followed to build a containerized serverless chatbot. The implementation for the proposed strategy and methodologies is underway and will be carried out in the future.

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our guides (Prof. Anisha BS and Dr. GS Mamatha) who gave us a lot of guidance while working on this paper, which also helped us in doing a lot of research and helped us learn and explore many new things.

REFERENCES

[1] Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, Larry Heck, "Dialogue Learning With Human Teaching And Feedback In End-To-End Trainable Task-Oriented Dialogue Systems", *arXiv:1804.06512 [cs.AI], Apr. 2018, pp. 2-15*

[2] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, Milica Gašić in "Multiwoz – A Large-Scale Multi-Domain Wizard-Of Oz Dataset For Task-Oriented Dialogue Modelling", *arXiv:1810.00278 [cs.CL], Sep 2018, pp. 12-20.*

[3] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, Jason Weston in "Learning From Dialogue After Deployment: Feed Yourself, Chatbot!", *arXiv:1901.05415 [cs.CL], Jan 2019, pp. 20-35.*

[4] Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio "An updated performance comparison of virtual machines and Linux containers", *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 25-30.*

[5] Yunchun Li and Yumeng Xia in "Auto-scaling web applications in hybrid cloud based on docker", *2016 5th International Conference on Computer Science and Network Technology (ICCSNT), pp. 23-30.*

[6] T. Combe, A. Martin and R. Di Pietro, "To Docker or Not to Docker: A Security Perspective," in *IEEE Cloud Computing, vol. 3, no. 5, pp. 54-62, Sept.-Oct. 2016, pp. 12-17.*

[7] Abdul-Kader, SA and Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems" in *International Journal of Advanced Computer Science and Applications, 6 (7). ISSN 2156-5570, pp. 15-27.*

[8] Ole-Christoffer Granmo, Ewa Luger and Petter Bae Brandtzaeg, "Chatbot Research and Design", in *Third International Workshop, CONVERSATIONS 2019, Amsterdam, The Netherlands, November 19–20,pp. 12-20.*

[9] Hulya Vural, Murat Koyuncu and Sinem Guney, "A Systematic Literature Review on Microservices", in *International Conference on Computational Science and Its Applications.,pp. 1-7.*

[10] Jianping Wu and Caidong Gu, "Architectural Patterns for Microservices: A Systematic Mapping Study" in *International Conference on Information Engineering and Computer Science, 2009, pp. 12-17.*

[11] D Merkel, "Docker: lightweight linux containers for consistent development and deployment" in *Linux journal, 2014, pp. 17-20.*

[12] A. M. Rahman, A. A. Mamun and A. Islam, "Programming challenges of chatbot: Current and future prospective," *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dhaka, 2017, *pp. 75-78*.

[13] H. Kang, M. Le and S. Tao," Container and microservice driven design for cloud infrastructure devOps*", in IEEE International Conference on Cloud Engineering (IC2E), Berlin, 2016, pp. 202-211.*

[14] R. Punjabi and R. Bajaj, "User stories to reality: A devOps approach for the cloud," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & communication technology (RTEICT), Bangaore, 2016, pp. 658-662.*

[15] C. Barna, H. Khazaei, M. Fokaefs and M. Litoiu, "Delivering Elastic Containerized Cloud Applications to Enable DevOps," in *2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Buenos Aires, 2017, pp. 65-75.*

[16] B. Kohli, T. Choudhury, S. Sharma and P. Kumar, "A Platform for Human-Chatbot Interaction Using Python," *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 439-444.*

[17] R. Ravi, "Intelligent Chatbot for Easy Web-Analytics Insights," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 2193-2195.*