



Comparative study of cloud platforms to develop a Chatbot

Amit Patil¹, K Marimuthu^{1*}, Nagaraja Rao A¹ and R Niranchana¹

¹School of Computer Science and Engineering, VIT University, Vellore-632014, India

*Corresponding author E-mail:marimuthume@gmail.com

Abstract

Before chatbots there were simply bots: The invention of a chatbot brought us to the new era of technology, the era of conversation service. A chatbot is a virtual person that can effectively talk to any human being with the help of interactive conversational skill. Now a days there are many cloud-based platforms available for developing and deploying the chatbot such as Microsoft bot framework, IBM Watson, Kore, AWS lambda, Microsoft Azure bot service, Chatfuel, Heroku and many more but all those techniques has some drawbacks such as built-in Artificial Intelligence, NLP, conversational service, programming etc. This paper represents the comparison between all cloud-based chatbot technologies with some constraint such as built-in AI, setup time, completion time, complexity etc. Finally, by the comparison, we will get to know that which cloud platform is efficient and suitable for developing chatbot.

Keywords: Artificial Intelligence; Chatbot techniques; Cloud Platform; NLP; Web hook;

1. Introduction

Since 2016, the world becomes digital with new the technologies. One of the best examples is "Chatbot". It is a computer program designed to interact with users via textual or auditory methods using artificial intelligence. We can also call it as a Personal digital assistant. Now we are living in an era where everything is available through the internet. We can get all type of information but this can be possible by typing for that the chatbots are invented. Depending on the application use such as shopping, customer services, food order, news updates, reservation etc. we can develop the chatbot. The popularity of messaging apps suggests people will happily talk to chatbots. Basically, chatbot requires chat interface or chat window which takes user inputs and provide a response message to the user[1].

The bots can be developed for entertainment or business purpose. Depending on the way we developed the bots, we can categorize the bot in two way one is Command-based bots and the other is smart bots. Command-based bots are manually programmed by a developer with the help of user inputs. The functionality of command-line bots is limited as they are not using the cognitive services to programmed the bots. Smart bots depend on artificial intelligence to interact with users. Instead of going through the predefined answer, smart bots predict response message based on the context and previous message.

This paper presents the introduction of cloud-platform which can be used to develop the chatbot. There are many cloud-platforms such as Microsoft Azure bot service, IBM Watson, Chatfuel, Heroku, Kore, AWS lambda etc. which provides bot services and built-in cognitive services which are easy to configure chatbot. All the cloud-platforms differs with their own features and functionalities that I will compare in analysis part. Azure Bot Service[2] provides an integrated environment in which we can deploy, build, test, and connect with

different channels that interact naturally wherever your users are interacting. With the IBM Watson Conversation service[3], we can create chatbot application that combines cognitive approaches build and train a chatbot using intents and entities and constructing dialog to simulate conversation. AWS Lambda[4] is not a Cloud Chat bot platform, but it uses lambda functions which facilitate easy creation and deployment of Chatbots. Kore[5] is a complete enterprise grade platform-as-a-service (PaaS) that enables companies and developers to design, create and deliver superior, highly intelligent, Natural Language Processing (NLP) enabled bots for use in many communication channels. Another chatbot service is Chatfuel[6] which is very simple to develop chatbot with no programming language skill is required. It can be easily integrated with Facebook and other social channels. Heroku[7] is cloud platform that enables developers to build, run, and test chatbot which supports built-in artificial intelligence as well as programming languages.

All these cloud platforms are used to develop the chat bot but in order to interact with bot, we require some interface. Users can chat with bots via text, e-mail, and popular messaging tools such as Facebook, telegram, Skype, any website or mobile app. For Facebook messenger, we need to create their own application and page then by using webhooks[8]. Figure 1. Shows Working of Chatbot which represents how chatbot actually works. The chatbot framework consists of three modules front end, back end and web hooks.

1.1. Front End

Front end can be any channel which provides the chat interface to interact with the bot. The popular messenger such as Facebook, Telegram, Skye, Slack and much more provide chat interface to the users to interact with bot user-friendly. There are some programming languages such as Node.js is one example offers its own real time

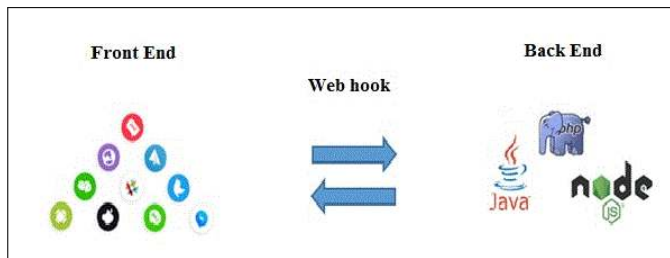


Figure 1: Working of Chatbot

chat interface.

1.2. Back End

Chatbots can be built on any cloud platform which may support programming language that allows you to make a web API. Programming languages such as Node.js or PHP, which are commonly used to build a chatbot but Java and Python provide inbuilt bot libraries as well. The backend is responsible for receiving user input, process the user message using some external APIs or Algorithms, and generate the response message.

1.3. Connecting the two

The connection between bot and the chat interface is done using Web Hook- URL based connection. A web hook conveys user message to other applications as it happens, meaning you get data immediately. If we want to integrate our bot to the Facebook messenger, initially we have to set up web hook then we can interact with bot using Facebook messenger.

1.4. Road map of this paper

This paper is divided as follows: in Section 2 we have written the literature survey on studies related to chatbot and cloud platform; in Section 3 we describe the methodology to develop chatbot using cloud platform and its architecture. In further section 4, we evaluate all the cloud platform using some results and finally ending with the conclusions and future work in Section 5.

2. Literature Survey

There are many techniques and other services such as Speech-to-Text, Text-to-Speech, and natural language processing etc. where the bot can be interactive. Kader et al.[9] presented the design techniques for developing interactive chatbots. They used Natural language processing techniques such as NLTK which can be used to analyze speech and make the bot response intelligent. They have done the survey of nine selected studies and also discussed the comparison between the chatbot design techniques.

The authors in [10] discussed the different chatbot strategies and also compared the conversion techniques such text-based conversion and speech based conversion. They also discussed some parameters which affect human-computer interaction quality in conversational systems which can be used to design web interface. Now-a-days many different chatbots are found through the web. Pereira and Coheur[11] described their own chatbot platform “Just.Chat” which can be used to process the information for developing the chatbots. They also discussed “Ed-gar” platform which is designed for answering natural language questions. Based on the filter such as Domain filter, Personal filter, and Blacklist Filter, they identified the interaction and created chatbots knowledge bases.

Cloud services are separating its application from its hardware and software dependencies. There are many Cloud service providers

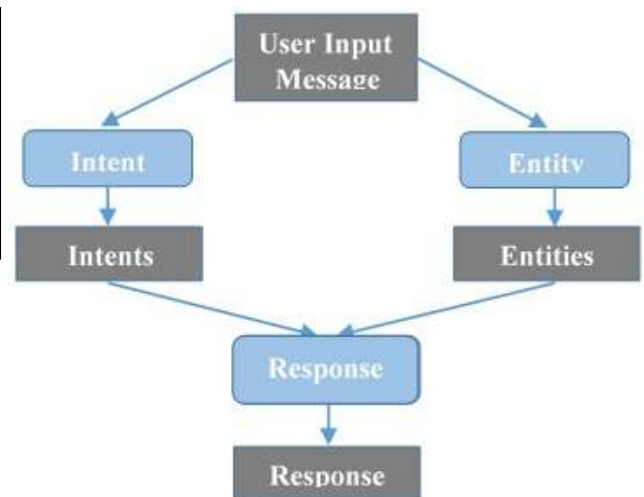


Figure 2: Chatbot Architecture

including Google, Microsoft and Amazon Web Services etc. Gandhi and Kumbharana[12] presented a comparison between Amazon Web Service and Microsoft Azure platform for choosing the cloud services. They have compared AWS and Microsoft Azure based on some parameter such as Base plan price, virtual CPU core, RAM, Disk space, IDE support, server OS type and much more. Selection of cloud service provider depends on the application requirement and the cloud services that are necessary to develop the application.

3. Methodology

With the recent increase in the popularity of chatbot, we need to think about the interaction between the user and the bot. We need to analyze user input message, and the bot response as well[13]. Figure 2 Shows the general chatbot architecture. The interaction begins with user input message what user is talking about. Then the User input message will be processed through two modules intent classification and entity recognition. Intent classification module checks the user input message and identifies the purpose user message. Based on the number of intents and context of the input message it identifies the intents. Entity recognition module recognizes user message structured and extracts the main keyword from the bits of information, for example, The Airline bot can extract city and date. Both Intent classification and entity recognition modules are very important to find out the intents and entities throughout the interaction between the user and a bot. Next important module in chatbot architecture is response generator. It uses some external APIs and algorithm to generate the response. The response generator use intent and entities, as well as the context of the conversation, extracted from the last user message.

Now-a-days many cloud-platforms provide the bot services where we can develop the bot and deploy to any one of the cloud. There are some cloud-platforms which provide different services apart from the bot service such as built-in artificial intelligence, Cognitive services etc. In this paper, we have used the Microsoft Azure cloud platform to develop the chatbot. Microsoft bot framework consists of Bot builder, bot Connector, and bot directory. It also has an emulator where we can test the working of the bot. Bot builder SDKs for Node.js, Net are available to build the bot. If we want our bot to be more interactive, we can incorporate Microsoft cognitive services such as Language Understanding Intelligence Service (LUIS) [14]. Figure 3 Shows Microsoft Bot Framework which describes how the users communicate[15, 16, 17, 18, 19] with the bot using the messenger apps. In Microsoft bot framework, the role of bot connector is very important. The responsibility of bot connector is to connect with the different channels some of which are Facebook, Skype,

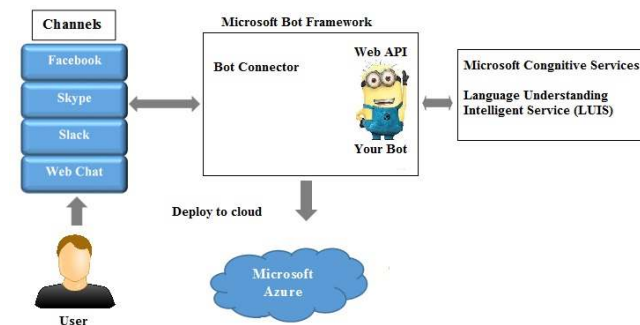


Figure 3: Microsoft Bot Framework

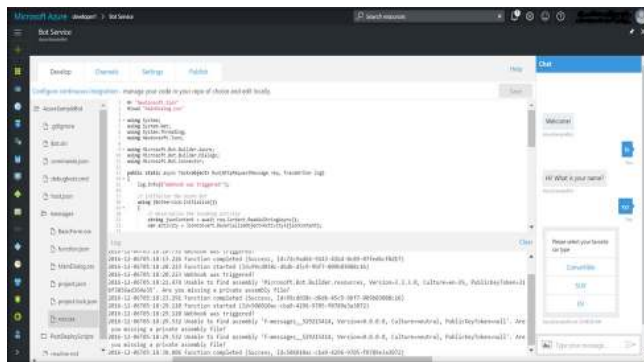


Figure 4: Microsoft Azure Integrated Environment

Skype, Slack, Web Chat etc. Once the bot is built, next job is to publish and deploy the bot to the Microsoft azure cloud platform. For that, it is mandatory to create the azure bot service in the azure cloud platform. Using the azure bot service, the bot is deployed to the cloud.

4. Result and analysis

As we have already discussed the supported cloud platform to build a chatbot in the First section. There are many advantages and disadvantages of chatbot cloud platforms based on their function-alities and features. In this section, we compare all the bots devel-oped in different cloud environments and their results. There are many cloud platforms, but here we discuss most commonly used platforms these includes Microsoft Azure bot service, IBM Watson, Heroku etc. We show the integrated environment as well as analysis table which are used to develop the chatbot based on user requirement. Figure 4. shows the integrated environment of Mi-crosoft azure where we can code, test, deploy, and publish the bot. Table 1 depicts the features, pros, and cons of Microsoft Azure platform.

4.1. Microsoft bot framework

Figure 5. Shows the integrated environment of Heroku cloud platform. It is also similar to Microsoft Azure, but it doesn't have in-built Artificial intelligent so it is that much popular than Microsoft Azure platform. Table 2 depicts the features, pros, and cons of Heroku platform where we can analysis the result of chatbot.

4.2. Heroku

Figure 6. Shows basis flow of IBM Watson conversion service. This is very simple to design the dialog, but the main disadvantage is we cannot design dynamic flow of conversions. The Watson Conversion service consists of Intents, Entities, and dialog through which we can build interactive smart bot. Table 3 depicts the features, pros,

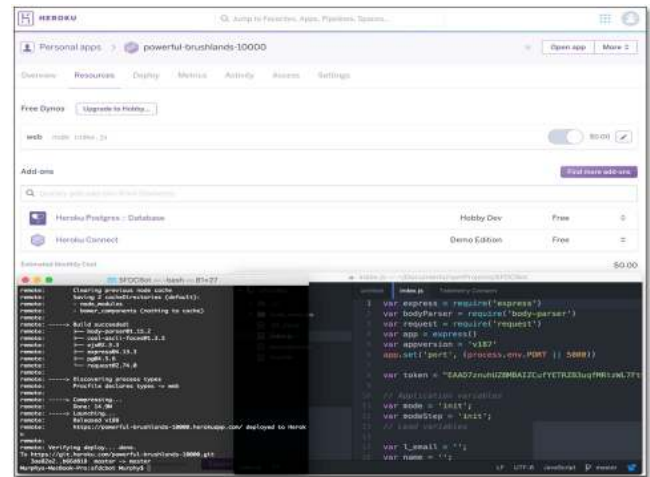


Figure 5: Heroku Bot environment

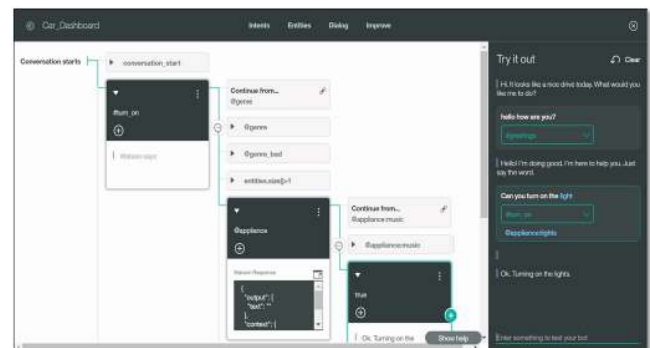


Figure 6: IBM Watson Bot Service

and cons of IBM Watson cloud platform.

4.3. IBM Watson

Figure 7. Shows the comparison between all the cloud platform by comparing some parameter some of they are AI Built-In, programming need, time to setup. Supported IDE etc. The below table also shows the pros and cons of all the cloud platform. Chatbot development is varied depends on who is going to develop the bot. If the developer is not from a coding background, he/she can choose KORE and Chatfuel platform to build the chatbot, but in order to build interactive chatbot, it should have in-built AI. Below table shows cloud platforms which have the built-in AI such as KORE, Microsoft bot framework, and IBM Watson. We can use different IDE to develop chatbot and some of the cloud platform has its own built-in IDE some of these are KORE, Chatfuel, Microsoft, Azure, AWS Lambda and IBM Watson. By this analysis table, anyone can choose the cloud platform in order to build and deploy the chatbot.

| | KORE | Chatfuel | Microsoft Bot Framework | Azure | HEROKU | AWS Lambda | IBM Watson |
|---------------------------------|-----------------------------|-----------------------|--------------------------|---------------------------|--|---|--|
| AI Built in Programming needed? | Yes No | No NO | Yes Yes | Yes Yes | No Yes | No Yes | Yes Yes |
| Complexity | High | Low | High | High | High | High | Medium |
| Time needed for setup | 10 min | 10 min | 1 hour | 15 min | 2 hours | 1 day | 4 hours |
| Pre-trials IDE | 8 hours Build in | 4 hours Build in | 8 hours Visual Studio | 8 hours Eclipse / Atom | 8 hours Eclipse / Atom | 16 hours Build in Eclipse / command line | 16 hours Build in Eclipse |
| Pro's | Extensive + all on platform | Plug'n'play | Most extensive | Integrated environment | Single development + deployment platform | Serverless deployment | High quality of interaction |
| Con's | Steep learning curve | Limited possibilities | Needs setup/ deployment | Only preview mode | Steep learning curve | Steep learning curve | Limited options to integrate with other services |

Figure 7: Companion Table for all Cloud Platforms

Table 1: Analysis table for Microsoft Azure cloud platform

| Channels | Programming Languages | Artificial Intelligenc | Integrates with | Pro's | Con's |
|--|-----------------------|------------------------|-------------------------|---|---|
| Slack Facebook Messenger Skype GroupMe Telegram Twilio | Node.Js,C# | Built in - LUIS | Any API via programming | - No IDE required Integrated environment - Runs on Azure serverless architecture (Azure Functions) - Well documented | - Runs currently in preview mode - Required Azure subscription |

Table 2: Analysis Table for Heroku Platform

| Channels | Programming Languages | Artificial Intelligenc | Integrates with | Pro's | Con's |
|---|---|------------------------|--|---|---|
| Slack Skype Web Mobile Email SMS | Node Ruby Java PHP Python Go Scala Clojure | None | SFDC as its part of Heroku ecosyste Any API via programming | - Secure (https) - Endpoint,Integrated with GIT so simple deployment., - Single platform for development & deployment | - Complex - Steep learning curve - Programming skills required. |

Table 3: Analysis table for IBM Watson

| Channels | Programming Languages | Artificial Intelligenc | Integrates with | Pro's | Con's |
|--|---|---|-------------------------|---|--|
| Facebook WeChat Telegram Kik Line Kakao | NodeJS Python Java Unity Android IOS | Built in Watson Conversation Other Watson Cognitive Services | Any API via programming | - Minimal number programming required - High Quality of Interaction - Proven solution | - Limited option to integrate various Watson services - Static Dialog |

5. Conclusion

In this paper, we introduced the chatbot concept and the different cloud platforms to develop chatbot. The developers, messenger apps as well as business become working together and building a new environment, because of the invention of chatbots. The chatbot can be developed for any purpose such as shopping, customer services, food order, news updates, reservation and much more. All the cloud platform discussed in this paper has different features and functionalities and based on these we have developed some efficient and interactive chatbot and got some results. By using the analysis and results, anyone can choose the cloud platform to build the chatbot. In future research, we will work on how to train the bot using built-in artificial intelligence so that the user will fill like they are taking with another human being. Also, we can think that how the bot will response dynamically without using context, intents or entities.

Acknowledgement

I would like to thank VIT University for letting us use the computer labs for our project work and also thankful to all our colleagues for their suggestions.

References

- [1] Kehoe, Ben, Sachin Patil, Pieter Abbeel, and Ken Goldberg, "A Survey of Research on Cloud Robotics and Automation", *IEEE Transactions on Automation Science and Engineering*, Vol.12, No.2, (2015), pp.398-409.
- [2] Creating your first bot-Docmentation-Bot Framework. [Online]. Available: <http://www.garypretty.co.uk/2016/07/14/creating-your-first-bot/>
- [3] Conversation-IBM Watson Developer Cloud. [Online]. Available:<https://www.ibm.com/watson/developercloud/conversation.html>
- [4] Create and Deploy a Chat Bot to AWS Lambda in Five Minutes-AWS Compute Blog. [Online]. Available:<https://aws.amazon.com/blogs/compute/>
- [5] Chatbot Intelligence-Kore. [Online]. Available:<https://kore.com/bots-platform/chatbot-intelligence/>
- [6] Create Chatbot for free. Best chat bot platform ChatFuel. [Online]. Available:<https://chatfuel.com/platforms>
- [7] Cloud Application Platform-Heroku. [Online]. Available:<https://www.heroku.com/>
- [8] Developer's Introduction To Chatbots-Tutorialzine. [Online]. Available:<http://tutorialzine.com/2016/11/introduction-to-chatbots/>
- [9] S. A. Abdul-kader and J. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems", *International Journal of Advanced Computer Science and Applications*, Vol.6, No.7, (2015), pp.72-80.
- [10] S. A. Abdul-kader and J. Woods, "A Survey on Web based Conversational BOT design", *Journal of Emerging Technologies and Innovative Research*, Vol.3, No.10, (2016), pp.96-99.
- [11] M. J. Pereira and L. Coheur, "Just . Chat - a platform for processing information to be used in chatbots", [Online]. Available:http://www.chatbots.org/ai_zone/viewthread/492/
- [12] V. A. Gandhi, C K Kumbharana, "Comparative study of Amazon EC2 and Microsoft Azure cloud architecture", *International Journal of Advanced Networking Applications*, ISSN 0975-0290, (2014), pp.117-123.
- [13] Chatbot Architecture. [Online]. Available:<https://hackernoon.com/chatbot-architecture-496f5bf820ed#.mtmd52u8i>
- [14] Bot Framework Overview-Docmentation-Bot Framework. [Online]. Available:<https://docs.botframework.com/en-us/>
- [15] S. Kumari, M. Karuppiyah, X. Li, F. Wu, A.K. Das and V. Odelu, "An enhanced and secure trust-extended authentication mechanism for vehicular ad-hoc networks", *Security and Communication Networks*, Vol.9, No.17, (2016), pp.4255-4271.
- [16] M. Karuppiyah, S. Kumari, X. Li, F. Wu, A.K. Das, M. K. Khan, R. Saravanan and S. Basu, "A dynamic id-based generic framework for anonymous authentication scheme for roaming service in global mobility networks", *Wireless Personal Communications*, Vol.93, No.2, (2016), pp.383-407.
- [17] M. Karuppiyah, "Remote user authentication scheme using smart card: a review", *International Journal of Internet Protocol Technology*, Vol.9, No.2-3, (2016), pp.107-120.
- [18] M. Karuppiyah, S. Kumari, A.K. Das, X. Li, F. Wu and S. Basu, "A secure lightweight authentication scheme with user anonymity for roaming service in ubiquitous networks", *Security and Communication Net-works*, Vol.9, No.17, (2016), pp.4192-4209.
- [19] X. Li, J. Niu, M. Karuppiyah, S. Kumari and F. Wu, "Secure and Efficient Two-Factor User Authentication Scheme with User Anonymity for Network Based E-Health Care Applications", *Journal of medical systems*, Vol.40, No.12, (2016), pp.268.