Develop a Simple Chatbot with API.AI

Kharis Isriyanto
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514064@std.stei.itb.ac.id

***Abstract*—Chatbot is one part of the technology that is still evolving. Using Natural Language Processing, which is part of Artificial Intelligence, the development is still taking place. To build a simple chatbot, we can use an NLP tools such as API.AI to save time and effort. The result we achieve is satisfying.**

Keywords—artificial intelligence; natural language proccesing; chatbot; API.AI

I. Introduction

The evolution of artificial intelligence has reached an amazing level. Now, machines can do something that require intelligence, which is maybe back in the past only human can do. For example there is a self-driving car called Waymo, developed by Google. It is autonomous, which means having the power for self-governance.[1] The autonomous car can decide what to do without any external intervention like user command. In video games, artificial intelligence also make a powerful impact to user experience. A more intelligent machine can make the game harder and make the player more challenged.

One of the most used artificial intelligence product in our daily life may be virtual personal assistant like Siri by Apple, Cortana by Microsoft, or Google Assistant. The similarity between those product is the usage of Natural Language Processing (NLP), which is a branch of artificial intelligence. It makes the machine understand what the user say and help them decide what to do next.

Beside those popular product as mentioned above, NLP has been used in many application to solve problem. It can be used to do a sentiment analysis from a sentence, extract information from a paragraph, or even chat. And now, there is so many chatbot developed, whether it is used for chatting only, like Mitsuku and Sim Simi, or for giving a service to the user.

There is so many benefit of chatbot. First, it can increase the user experience for a product. Imagine if you can ask to turn off the lamp to machine instead of stand up and walk to the switch. Second, it will give more human feeling to the user. The chatbot will talk to user with a language the user understand. So it will like talking or chatting with a friend.

II. Theoretical Basis

A. Artificial Intelligence

Artificial Intelligence (AI) according to Oxford Dictionaries is "the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages".[2] Where as Russell (2009) categorized AI to four part based on the approach taken, that are thinking humanly, thinking rationally, acting humanly, and acting rationally.[3]

So, an artificial intelligent agent should perform something based on one in that four approach. For example a self-driving car can belong to AI because it is acting rationally, it acts based on a computation and tried to achieve the best outcome. A machine that can communicate in English is an artificial intelligent agent because it can acting humanly.

B. Natural Language Processing in API.AI

Natural Language Processing is part of artificial intelligence. There are so many techniques and approach to process a natural language, such as techniques to define language models, to classify text, to retrieve information, and so on. But we will focus on the Natural Language Processing technique provided by API.AI tools because we will build chatbot with the help from API.AI

The most used feature in API.AI is NLU or Natural Language Understanding.[4] It is part of Natural Language Processing. NLU aims to transform a word sequence, called query, as input data, into actionable data. This is used by API.AI to mapping sentence user says into something called intent.[4] For example if someone say hello the machine can conclude the intent is greeting. But if someone say hi, or hey, the machine still have to conclude the intent is greeting because it says the same thing and have a similar goal. Another example is if user says "I want a burger" or "I want a rice", that queries have to be mapped to a same intent. This intent will help the machine decide what to do next and give response. For example if someone greet the bot, the bot will replying greetings too. Intents have to be defined by developers, and developers have to train the machine.

To help the machine classify query into intent, API.AI has a feature called entities.[4] This entitiy are used to help machine extracting parameter values from natural language inputs. There are system defined entities we can use such as time, date, location, number, etc. So if user say "How's the weather today?" and "How's the weather tomorrow?", API.AI can map those query into same intent. We can make a new entity such as food entity which contain hamburger, rice, spaghetti, or hotdog so we can use them to classify an intent.

There is still a lot of feature in API.AI beside intent, entity, and training, such as action to make the app do something beside giving response only. The other is context that let machine maintain a conversation. But for this project we only use those three feature.

III. Implementation

Developing a Natural Language Processing algorithm takes a lot of time and effort to learn. So, for the NLP part of our chatbot, we will use an NLP tools. There are a lot of tools available for developers to use, starting from an open source software to a paid software. We will use a freeware tools called API.AI. It is a tools for developers to define the behaviour of the bot if it getting a message from user. We don't have to code, instead we will use the web interface of the app to define the bot's behaviour.

For the front end part, we will use LINE Messenger. LINE provide a messaging API (Application Programming Interface) for developers. We can develop a bot that interacts with users in LINE chats.[5] To use this API we need to make a server that will process all the coming messages. When the server get a message it will redirect the message to API.AI. API.AI will give a response back to the server based on the rules. Then the server will give the response back to the user.

In summary, we will build a system like the one in figure 1. User will give message to our bot in LINE Messenger. API.AI in the back-end system will process it with Natural Language Processing and give the response based on the rules we have decide.

We will limit the chatbot functionalities to reply greetings, answer a weather, and handle food order. But the bot only reply the user and not really doing something like order food.

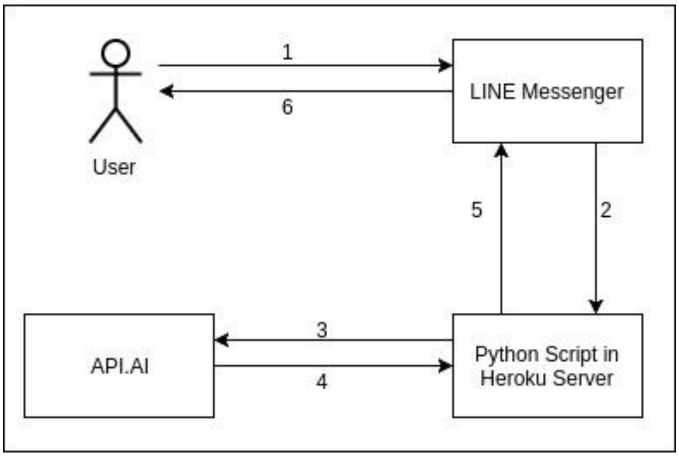
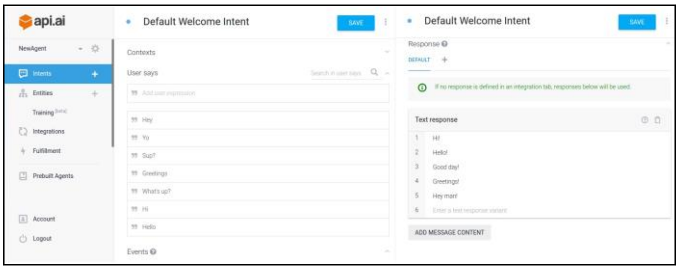


Figure 1. Message flow of the chatbot system

A. API.AI

First, we have to create a new app in API.AI. Next, we will define the intent for our chatbot. API.AI has defined two default intent, that is default welcome intent and default fallback intent. The default fallback intent will be used when API.AI fail to recognize the intent in a sentence. It will send a response asking user to say that again. API.AI provide some default response like "I didn't get that. Can you say it again?" or "I missed that." We can add as many response more as we like.

The default welcome intent has to be defined. For better result, list all the possibility a user greet the chatbot. We add the usual greeting word like "Hello", "Hi", or more casual greeting like "What's up?". Also add the bot response. API.AI has defined some response like "Hi!" and "Hello!", but add more response as you like.



Default Welcome Intent

Contexts

User says

Add user expression

Hey

Yo

Sup?

Greetings

What's up?

Hi

Hello

Events

Default Welcome Intent

Response

If no response is defined in an integration tab, responses below will be used.

Text response

1 Hi!

2 Hello!

3 Good day!

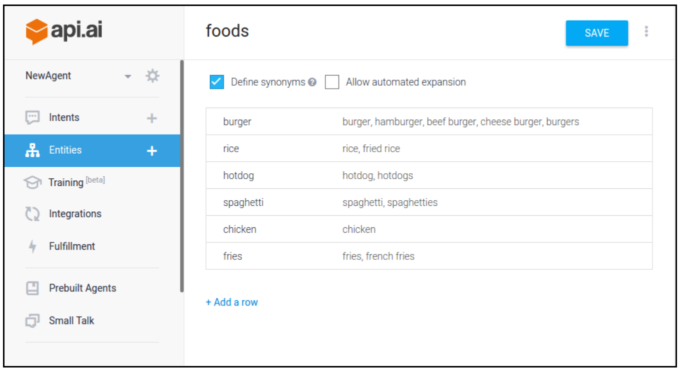
4 Greetings!

5 Hey man!

6 Enter a text response variant

Figure 2. Default welcome intent

We will make another intent to let the user ask the weather and temperature in someday. This intent will use a time entity. Also we will randomize the result between clear, cloudy, or rainy, because we don't know the actual weather. Next we will make a new entity called foods, which contains some foods so we can make intent to handle food order. Notice in figure 3 that we can define synonims, so if user say burgers or hamburgers, machine will perceive that as burger.



foods

☑ Define synonyms ☐ Allow automated expansion

burger	burger, hamburger, beef burger, cheese burger, burgers
rice	rice, fried rice
hotdog	hotdog, hotdogs
spaghetti	spaghetti, spaghetties
chicken	chicken
fries	fries, french fries

Figure 3. Entity of foods

B. LINE Messenger API

To make our bot in LINE Messenger we have to make a LINE@ account. To use LINE@ we have to pay based on our chosen plan, but we can use a free developer trial account. This account cannot have more than 51 friends, but it is okay because we don't need that much friend for our bot.

We have to download line-bot-sdk for Python, an SDK (Software Development Kit) provided by LINE. This Python script is responsible to run our server. It will handle the request from LINE and get the message body. This message body, which contain message the user send will be sent to Wit.ai.

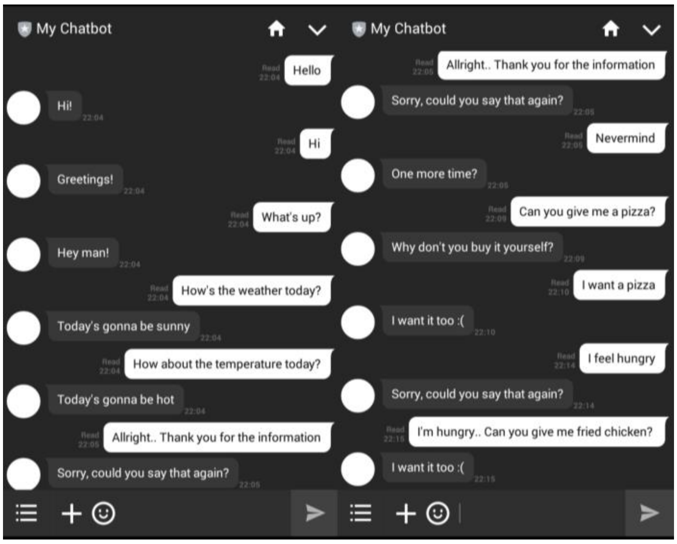
Then this server will get the response and reply the user with this response.

The main function we must have in our Python script is app routing, message handler, requesting to API.AI. App routing makes the coming request to the server will be handled by one function that is designed to handle that request. Message handler will handle the message and also handle the reply back to the user. Request to API.AI function will send the message to API.AI and handle the request.

To deploy our server, we use a cloud platform named Heroku. Heroku enables developers to deploy their apps without worrying about the infrastructure.[6] So, this Python script we have written will be deployed in Heroku. We must set the webhook in our bot account to the URL in Heroku so that every message we get in LINE Messenger will be sent to the Heroku app we deploy.

C. Result

The bot we create is working. You can see one of the conversation in figure 4 below. As we can see, the bot reply as we defined before, except for the message that never defined before like “nevermind”, or “I feel hungry”. API.AI will randomize the response so user will not be bored and stay engaged. If we add some more functionality to the app, it can do more things, such as retrieve the actual weather from an external service, or order food from another application.



My Chatbot

Read 22:04 Hello

Hi! 22:04

Read 22:04 Hi

Greetings! 22:04

Read 22:04 What's up?

Hey man! 22:04

Read 22:04 How's the weather today?

Today's gonna be sunny 22:04

Read 22:04 How about the temperature today?

Today's gonna be hot 22:04

Read 22:05 Allright.. Thank you for the information

Sorry, could you say that again? 22:05

My Chatbot

Read 22:05 Allright.. Thank you for the information

Sorry, could you say that again? 22:05

Read 22:05 Nevermind

One more time? 22:05

Read 22:09 Can you give me a pizza?

Why don't you buy it yourself? 22:09

Read 22:10 I want a pizza

I want it too :(22:10

Read 22:14 I feel hungry

Sorry, could you say that again? 22:14

Read 22:15 I'm hungry.. Can you give me fried chicken?

I want it too :(22:15

Figure 4. Conversation example

IV. Conclusion

Building a chatbot would be difficult if we develop the algorithm and the application from scratch. Natural Language Processing is challenging that it would take time and a lot of effort to achieve a good result.

A tools like API.AI help developers to develop an NLP app within hours without develop the algorithm from scratch. Developers just have to define the words to intent, and decide what to do for the bot. Of course if we use tools, it is limited to the tools’ functionalities.

Acknowledgment

The author give thanks to God for His grace, this paper can be done in time. Author give thanks to Dr. Ir. Rinaldi Munir, M.T., Dr. Eng. Ayu Purwarianti, S.T., M.T., and Dessi Puji Lestari S.T., M.Eng., Ph.D. as lecturers in IF3280 Socio-informatics and Profesionalism course. Author also give thanks to all the developers who develop tools and services used in this project, those are API.AI, LINE Messenger API, and Heroku. Also author would like to thank friend and family for support in this project.

References

[1] Antsaklis, Panos J, Passino, Kevin M., Wang, S.J., (1991)."An Introduction to Autonomous Control Systems".*IEEE Control Systems.*

[2] Definition of Artificial Intelligence in Oxford Dictionaries, https://en.oxforddictionaries.com/definition/artificial_intelligence, accessed on May 5th 2017

[3] Russell, Stuart J.;Norvig, Peter(2009).*Artificial Intelligence: A Modern Approach*(3rd ed.). Upper Saddle River, New Jersey: Prentice Hall.

[4] Documentation of API.AI, https://docs.api.ai/docs/, accessed on May 5th 2017

[5] Documentation of LINE Messenger API, https://devdocs.line.me/en/, accessed on May 5th 2017

[6] What is Heroku, https://www.heroku.com/what, accessed on May 5th 2017

Statement

I hereby declare that the paper is my own writing, not an adaptation, nor translation from another person’s paper, and not a form of plagiarism.

Bandung, May 5th 2017

Kharis Isriyanto 13514064