IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Developing Facebook Chatbot Based on Deep Learning Using RASA Framework for University Enquiries

To cite this article: Yurio Windiatmoko *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1077** 012060

Developing Facebook Chatbot Based on Deep Learning Using RASA Framework for University Enquiries

Yurio Windiatmoko[1], Ridho Rahmadi[2], Ahmad Fathan Hidayatullah[3]

Department of Informatics, Universitas Islam Indonesia, Yogyakarta, Indonesia

E-mail: [1]yurio.windiatmoko@students.uii.ac.id, [2]ridho.rahmadi@uii.ac.id , [3]fathan@uii.ac.id

Abstract. Smart systems for universities powered by artificial intelligence have been massively developed to help humans in various tasks. The chatbot concept is not something new in today's society which is developing with recent technology. College students or candidates of college students often need actual information like asking for something to customer service, especially during this pandemic, when it is difficult to have an immediate face-to-face meeting. Chatbots are functionally helping in several things such as curriculum information, admission for new students, schedule info for any lecture courses, students grade information, and some adding features for Muslim worships schedule, also weather forecast information. This Chatbot is developed by deep learning models, which was adopted by an artificial intelligence model that replicates human intelligence with some specific training schemes. This kind of deep learning is based on RNN which has some specific memory savings scheme for the deep learning model, specifically this chatbot using LSTM which already integrates by RASA framework. LSTM is also known as Long Short Term Memory which efficiently saves some required memory but will remove some memory that is not needed. This Chatbot uses the Facebook platform because the Facebook users have already reached up to 60.8% of its entire population in Indonesia.

Keywords. Chatbot; Deep Learning; Chatbot Framework; RNN; RASA; LSTM

1. Introduction

Chatbot is the easiest system to access by any users; it is an artificial intelligence technology for communicating through natural language with humans. Chatbot makes it possible to understand human language through natural language processing. Language is a social and cultural reciprocal for a natural protocol. It is time for computer machines to add features to understand humans along with the increased performance of the recent computer machine hardware [1].

College students or candidates of college students often need actual information, e.g., asking for something to customer service, especially during this pandemic, when it is difficult to have an immediate face-to-face meeting. Therefore, a chatbot is needed for answering questions quickly, correctly, and is available 24 hours [2]. Functionally, this Chatbot is helping in several things such as curriculum information, admission for new students, schedule info for any lecture courses, students grade information, and some adding features for Muslim worships schedule, also weather forecast information.

To help solve the problems above, we propose a chatbot using deep learning models. Deep learning consists of high-level abstract modeling algorithms on data using transformation of nonlinear functions arranged by many layers and deeply [3]. Deep learning is also a branch of machine learning, inspired by the neural network of the human brain which is represented by layers on the neural network [4]. The deep learning method used in this case is RNN based called LSTM (Long-Short Term Memory) which has some specific memory savings scheme for the deep learning model. LSTM is able to efficiently save some required memory and also will remove some memory that is not needed. We also use a framework called RASA to build this chatbot application which already integrates LSTM as the deep learning method. The implementation of deep learning on chatbot requires the role of the chatbot framework which is for unifying and pipelining the deep learning model. The sense of a chatbot framework is to implement its conversation flow. RASA works on two main procedures namely RASA NLU and RASA Core. RASA is a library of tools from the Python programming language that is open source for building conversational software. The aim and philosophy of the initiators of the RASA chatbot framework are to create machine learning-based dialogue management that provides ease of use in terms of implementation, as well as bootstrapping, by starting the stages from existing sources to create something more complex and in a more efficient way, even with minimal initial training data [5].

2. Literature Review

There are two main approaches to generate a chatbot response. If the chatbot uses a traditional approach then it is hard-coded rule-based templates and rules that process for responses. However, nowadays many new and interesting approaches allow deep learning approaches to emerging. Neural network models are trained with several data to study the process of generating responses that are grammatically relevant and in terms of user speech intent [6].

These are some previous works related to chatting systems or chatbot for universities. Online Chatting System for college inquiries using a knowledgeable database by Bathe, Malusare, and Kolpe is doing pattern matching to perform information retrieval on chatbots. It has not already used the deep learning approach, so that is still too rigid by rules for users to make inquiries. But, detailed working steps in this research are very good, there such a UML, and various kinds of process diagrams [7]. Erasmus the AI chatbot, Erasmus is a chatbot on Facebook which is used to answer the queries related to the college information. Designing end-to-end systems using cloud services, starting from api.ai (Dialogflow), Mlab (MongoDB cloud), IBM Bluemix (webhook API), scrapper import.io to minimize the scripting code process, but what's happening here, it is all because there are too many diverse cloud services, there is quite a long latency between the services [8]. Then, Eaglebot a Chatbot Based Multi-Tier Question Answering System for Retrieving Answers from Heterogeneous Sources Using BERT. It is a scalable chatbot system using 3 route selection methods with the main framework using Dialogflow, then several completion methods from document retrieval and document reader [9]. Eaglebot which serves to answer various questions commonly asked by students in the university domain. However, the Eaglebot application still has some limitations by Dialogflow chatbot framework which has some request limits, if it wants to unlimit requests you need to subscribe for the member privileged.

Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network [10] by Anran Jiao was comparing performance between RASA NLU stack with Neural Network Classifier and Entity Extractor from scratch. This research was told that the RASA NLU method still superior to extract all the entities and classify user speech intent. It's also quite comprehensive research but this research only depends on a free API for response completion and not using its own database. Compared to all those previous works, this research will be more reliable by using deep learning for chatbot natural language understanding and combining free API with it's own database for response completion. So it won't need for pattern matching, not using cloud provider service-based, and not only depends on a free API service so that customization will be easier, scalable and no limit request for user interaction.

Facebook users in Indonesia August 2020. There were 166.500.000 Facebook users in Indonesia in August 2020, which accounted for 60.8% of its entire population. The majority of them were men - 54.6%. People aged 25 to 34 were the largest user group 59.000.000. The highest difference between men and women occurs within people aged 25 to 34, where men lead by 9.000.000. So that this Chatbot uses the Facebook platform as the user interaction medium.

3. Data and Methods

All the processes in this research are following this diagram process flow below.

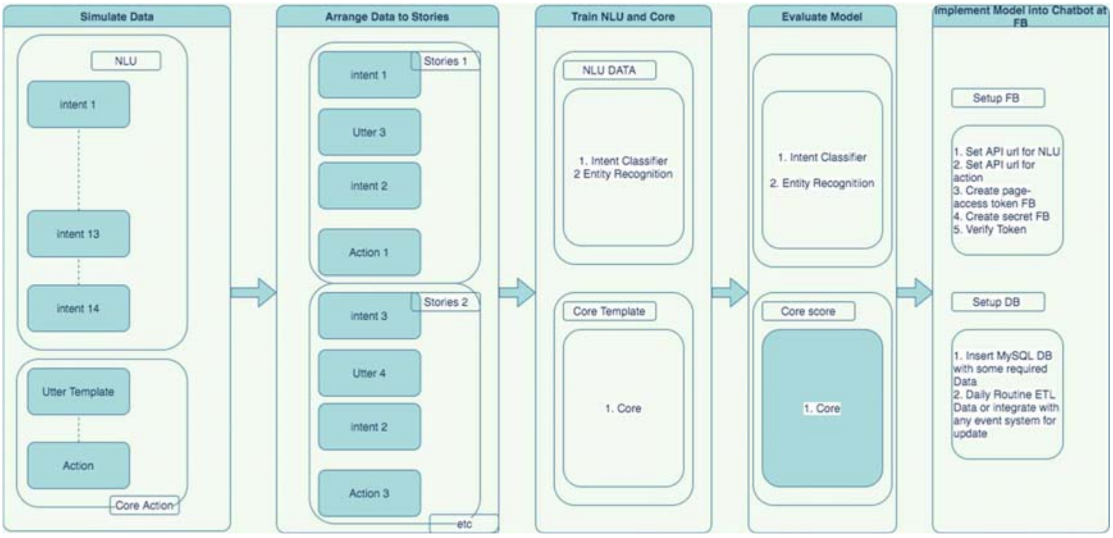


Figure 1. Workflow chart chatbot

Figure 1 describes flow and methods of the research process carried out. Beginning with simulating the data based on common dialogue interaction along with grouping for each sentence into the intent topic, utter chatbot response template, and action. Especially in the intent topic which has been proposed as the featured chatbot also needs defining Entities there. The difference between utter and action here is basically that utter is just a sentence with an expression of the general response, whereas action is utilized when it's response needs some completion based on a database or API, also it needs wrapping-up as the class actions on RASA. At this stage, various data previously seen and considered from general reference schemes or conversation scenarios for each chatbot feature such as communication flow on student questions about the curriculum, admission for new students, schedule for lectures, student grade, Muslim Worships schedule, and weather forecast also adding up with some common dialogue intent like greetings and so on. Then, arrange those intent, utter and action in such a way that it corresponds to the flow of stories and common entities in each conversation sentence. After all the data is already set up, then train those data using RASA Trainer. Check and recheck after training was finished by evaluating the trained model. If the evaluation score obtained has reached the required expectations then just integrate that model with credential setup configuration after the Facebook developer account has been created.

3.1. Framework

Chatbot framework is used for Bootstrapping all the chatbot engine modules. Especially for the Machine Learning Models. Below is the Architecture flow of Rasa framework used.

The architecture or dialogue management flow used in this research approach is as shown in Figure 2 which is the architecture of the chatbot stack framework itself. The first stage of the message is received and forwarded to the interpreter, namely Rasa NLU to extract intents, entities, and other structured information. Both interpreter or tracker is tracking, detecting, and maintain the status of the conversation context through the message notifications it has received. The third policy or policy manager receives context status from the tracker. The fourth policy or policymakers choose which action will be taken next. The five actions or actions are recorded by the tracker. These six actions are executed by sending a message to the user. Seventh, if the action that has been executed is ignored or ignored by the user at a certain time, the process returns to the third step.

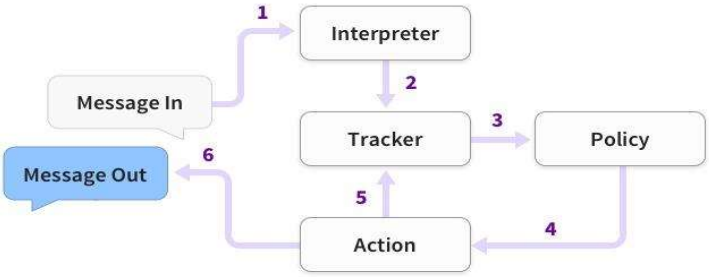


Figure 2. RASA stack framework

3.2. Methods For Training

Figure 3 depicts the intent entity training flow. Our chatbot system has 2 modules and each module is trained separately, those modules are NLU and Dialogue Policy or Dialogue Management. NLU or the interpreter comprises two component models inside. Those are Intent Classifier and Entity Extractor.

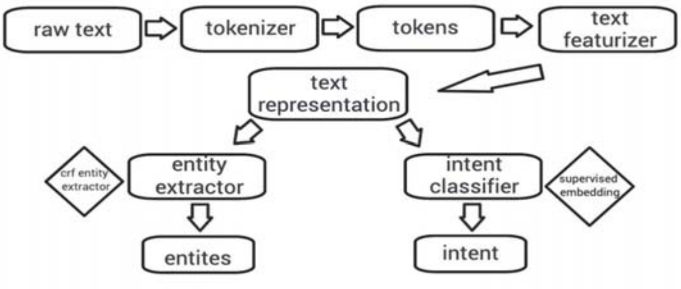


Figure 3. Intent entity training flow

Specifically, this chatbot used Supervised Embeddings as Intent Classifier, which is also known as Embedding Intent Classifier. The Embedding Intent Classifier embeds user inputs and intent labels into the same space. Supervised embeddings are trained by maximizing similarity between them. This algorithm is based on StarSpace [11]. However, in this implementation, the loss function is slightly different and additional hidden layers are added together with dropout. This algorithm also provides similarity rankings of the labels.

The training configuration to build this intent classifier model is 300 for epochs, then for the hidden layers sizes at feed forward neural networks are 256 at h1, 128 at h2, and 14 at output neural network label intent class. Embedding dimension used for training is 20, this is multiple embedding layers inside the model architecture. For example, vector of '__CLS__' token and it's intent is passed on to an embedding layer before they are compared and the loss is calculated. config for batch size is (64, 256) which the initial and final value for batch sizes. Batch size will be linearly increased for each epoch. Learning rate is set as 0.001 for this training config.

Entity Extractor used for this chatbot is crf entity extraction [12] which method is suitable for custom entities extractor. It's component implements a conditional random field (CRF) to do named entity recognition. CRFs can be thought of as an undirected Markov chain where the time steps are words and the states are entity classes. It's features are from the words like capitalization, POS tagging, etc. Those Features are give probabilities to certain entity classes, as are transitions between neighbouring entity tags, the most likely set of tags is then calculated and returned. Training configuration for CRFEntityExtractor has some list of features to use. However, it can be set and chosen by any requirements. Specifically this chatbot uses a list of features in such 'lowercase' to check whether the token is lower case, 'uppercase' to check whether the token is upper case, 'title' to check whether the token starts with an uppercase character and all remaining characters are lowercase, 'digit' to checks whether the token contains just digits, 'prefix5' to take the first five characters of the token, 'prefix2' to take the first two characters of the token, 'suffix5' to take the last five characters of the token, 'suffix3' to take the last three characters of the token, 'suffix2' to take the last two characters of the token, 'suffix1' to take the last character of the token, 'bias' to add an additional "bias" feature to the list of features.

As the featurizer is moving over the tokens in a user message with a sliding window, it defines features for previous tokens, the current token, and the next tokens in the sliding window. so the features are as [before, token, after] the array.

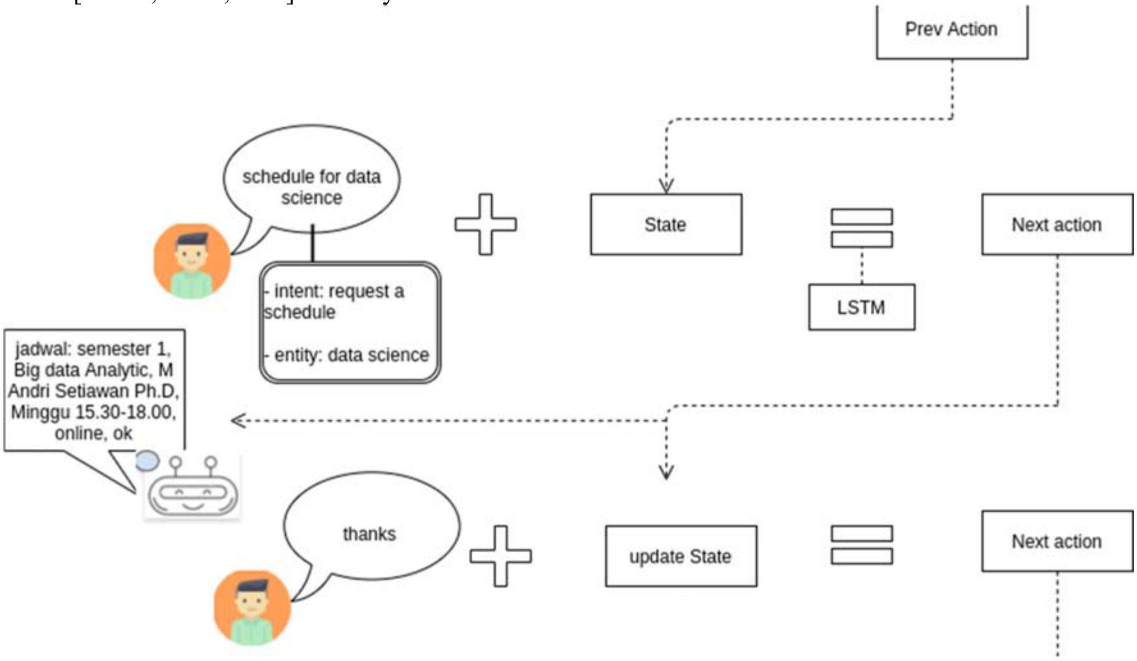

Prev Action
schedule for data science
State
Next action
- intent: request a schedule
- entity: data science
LSTM
jadwal: semester 1, Big data Analytic, M Andri Setiawan Ph.D, Minggu 15.30-18.00, online, ok
thanks
update State
Next action

Figure 4. Dialogue management component flow

As for the Dialogue Policy or Dialogue Management, it works by creating training data from the stories and training a model on that data. Stories are like some scenario or scheme from the common topic in any chat flow that we design or we expected. Chat Response works by creating training data from the stories and training a model on that data as described in Figure 4. Dialogue Policie decides which action to take at every step in the conversation. There are different policies to choose from, and it includes multiple policies in a single Agent or chatbot. At every turn, the policy which predicts the next action with the highest confidence will be used.

This chatbot already configured the policies by memoization Policy and neural network Policy. The execution of the policy will be based on the priority; highest priority policy will be executed first and so on. Priority is calculated on the basis of confidence score of each policy, which has a higher score given higher priority. The MemoizationPolicy just memorizes the conversations in training data. It predicts the next action with confidence 1.0 if this exact conversation exists in the training data, otherwise it predicts None with confidence 0.0. A Neural Network Policy implemented in Keras to select the next action. The default architecture is based on an LSTM [13]. It is configured with masking layer (input, 5, 32), lstm units (32) dropout=0.2, dense layer (19) then 100 training epochs.

3.3. Connector Module Platform

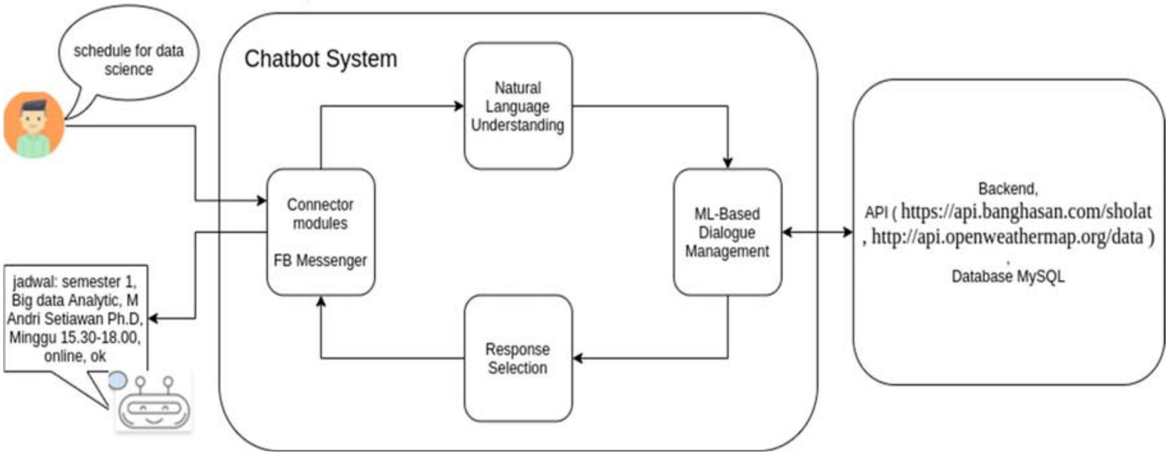

schedule for data science
Chatbot System
Natural Language Understanding
Connector modules
FB Messenger
ML-Based Dialogue Management
Response Selection
jadwal: semester 1, Big data Analytic, M Andri Setiawan Ph.D, Minggu 15.30-18.00, online, ok
Backend,
API (https://api.banghasan.com/sholat , http://api.openweathermap.org/data)
Database MySQL

Figure 5. Flow of chat system

Figure 5 illustrates the Facebook Messenger as a connector module platform, first it needs to set up a facebook page and app to get credentials to connect to Facebook Messenger. Once already have all of it, add these to the credentials.yml file. Then set up a Webhook and select at least the messaging and messaging_postback subscriptions. Insert the callback URL which will look like https://<HOST>/webhooks/facebook/webhook. Insert and Verify Token which has to match the verify entry in the credentials.yml file. So then, webhooks allows the chatbot to receive real-time HTTP notifications of changes to specific objects and postback to messaging.

4. Result and Discussion

This chatbot development research that uses deep learning models, it is expected to provide the best performance for chatbots systems through this deep learning model. Furthermore, the representation results of the performance evaluation of the deep learning model on the chatbot, namely intent classification and entity extraction at NLU and Dialogue Policy on the dialogue management chatbot system are measured and evaluated by precision score, recall score and F1-score [14].

Precision may be defined as the probability that an object is relevant given that it is returned by the system or here as the model predicted, while the recall is the probability that a relevant object is returned based on ground truth of each label data. Then, F1-score is needed for seeking a balance between precision and recall.

$$\mathit{Precision} = \frac{\mathit{True\ Class}}{\mathit{True\ Class} + \mathit{False\ Predicted\ Class}} = \frac{\mathit{True\ Class}}{\mathit{Total\ Predicted\ Class}} \tag{1}$$

$$\mathit{Recall} = \frac{\mathit{True\ Class}}{\mathit{True\ Class} + \mathit{False\ Predicted\ Oth\quad Class}} = \frac{\mathit{True\ Class}}{\mathit{Total\ Actual\ Class}} \tag{2}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Immediately, that precision shows how precise or accurate the model is out of those predicted Class, how many of them are actual Class. Recall calculates how many of the Actual Class our model captures through labeling it as those Class. Ultimately, F1-score is utilized to find a balance between precision and recall.

Table 1. Metrics evaluation intent classifier 1

	intent agreed	intent requests a schedule	intent goodbye	intent Muslim greeting	mood unhappy	intent only request a schedule	intent weather forecast	intent request a worship schedule
precision	1	1	1	0.75	1	1	1	1
recall	1	1	1	1	1	1	1	1
f1-score	1	1	1	0.85	1	1	1	1
data count	6	100	3	3	3	4	5	11
confused with	-	-	-	-	-	-	-	-

Table 1 shows the metrics evaluation intent for classifier 1. Here what needs to be underlined is the Intent Muslim greeting class where the model can predict accurately to the amount of truth data for that class which the recall score is 1, but the model still predicts the intent Muslim greeting in other intent classes actual which the precision score is 0.75 namely intent greeting class.

Table 2. Metrics evaluation intent classifier 2

	mood happy	intent greeting	intent thanks	intent request grade score	intent refuse	challenge bot	avg	total
precision	1	1	1	1	1	1	0.982	13.75
recall	1	0.85	1	1	1	1	0.98	13.85
f1-score	1	0.92	1	1	1	1	0.98	13.77
data count	4	7	2	10	3	4	11.78	165
confused with	-	'intent Muslim greeting: 1	-	-	-	-	-	-

In table 2, we describe the metrics evaluation for intent classifier 2. While for the intent greeting itself, here is get a 1 precision score where the Intent greeting is sometimes not detected by the model on the contrary intent Muslim greeting is even detected because it uses the same word phrase, which is 'assalamu alaikum' but this is not a problem because the chatbot template utter will answer the same thing as greeting utter.

The table 3 provides a detailed explanation of the precision and recall calculations of the two classes that are often misclassified. It can be seen from table 3 that the precision equals to 1. It shows that our chatbot system is able to detect the intent greeting precisely. As for the intent muslim greeting, the precision is 0.75.

Table 3. Confusion matrix between intent muslim greeting and intent greeting

		predicted			
		intent Muslim greeting	intent greeting	data count	recall
actual	intent Muslim greeting	3	0	3	1
	intent greeting	1	6	7	0.85
	total predicted	4	6	f1-score	
	precision	0.75	1	0.85	0.92

Table 4. Metrics evaluation entity recognition

	name	concentration study program	city	NIM	avg	total
precision	1	1	1	1	1	4
recall	1	1	1	1	1	4
f1-score	1	1	1	1	1	4
data count	10	170	12	3	48.75	195

In table 4, the entity class metrics evaluation table has shown the best performance from the model to be able for extracting the entity labels.

Table 5 illustrates the dialogue policy performance which has shown the best performance of the model in making decisions for the chatbot response. Student conversation flow requesting a lecture schedule from the chatbot has the following process. First, the chat is started with the greetings made by students to the chatbot, and the chatbot detects the greeting intent from users and replies with utter replies to the greeting. Furthermore, students go to the point to ask about their lecture schedule, and Chatbot detects it as an intent to request a schedule from users, then replies with an utter asking about the concentration of the student's study program. Then the student tells the chatbot the concentration of his study program, namely 'fd' or digital forensics, and the chatbot considers this to be the concentration entity of the study program and continues the query to MySQL DB as a completion to get lecture schedule data.

As for the alternative, the process of requesting schedule also may ask directly for a specific study program concentration schedule, for example, ‘schedule ds’ and the chatbot will detect the intent to request the schedule as well as to detect the concentration entity of the study program ‘data science’. And so chatbot continues the query to MySQL DB as a completion to get lecture schedule data.

Figure 6 shows an example of conversation from the student who asks about grade score. This chat flow is about requesting a grade score is almost the same as requesting lecture schedule flow, but only with different intents and entities, also different actions to show a list of grade scores by continuing the completion query to MySQL DB as a completion to get lecture schedule data. There is also a conversation flow of requesting the Muslim worship schedule. Beginning by asking directly for a specific intent about asking the Muslim worship schedule. Then the chatbot asking about a specific location as an entity, after that continuing the completion query directly to API https://api.banghasan.com/sholat/format/json/jadwal/kota/{city}/tanggal/{date.now}[1].

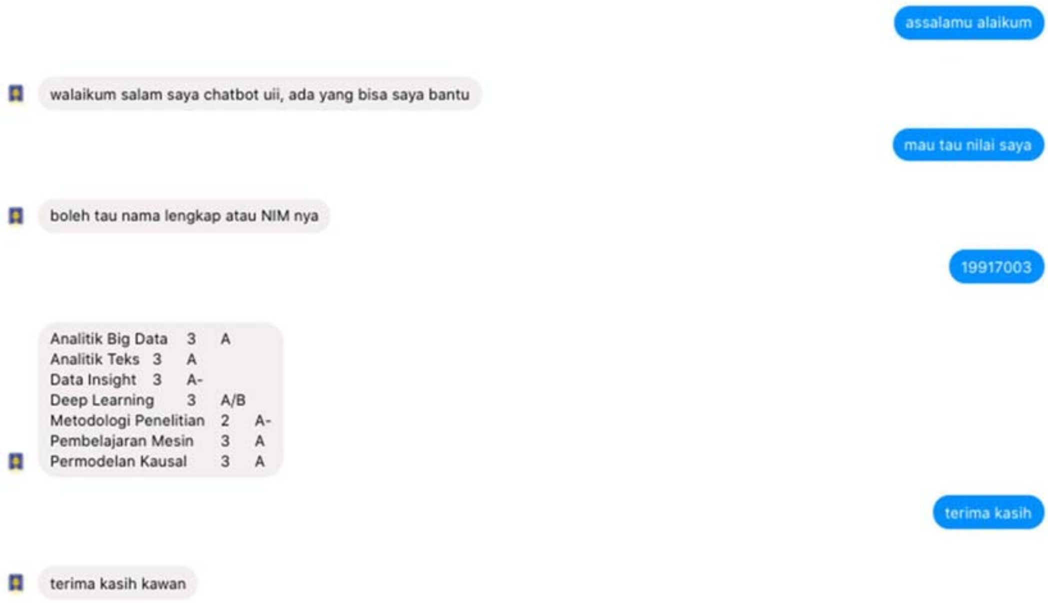
Table 5. Confusion matrix dialogue policy decision

	action schedule list	action listen	utter did it help	utter good bye	utter salam replies	utter say hello	utter cheers	utter im bot	utter happ y	utter asked concentrat ion study program	utter thank you
action schedule list	4	0	0	0	0	0	0	0	0	0	0
action listen	0	27	0	0	0	0	0	0	0	0	0
utter did it help	0	0	2	0	0	0	0	0	0	0	0
utter goodbye	0	0	0	2	0	0	0	0	0	0	0
utter salam replies	0	0	0	0	3	0	0	0	0	0	0
utter say hello	0	0	0	0	0	6	0	0	0	0	0
utter cheers	0	0	0	0	0	0	2	0	0	0	0
utter im bot	0	0	0	0	0	0	0	1	0	0	0
utter happy	0	0	0	0	0	0	0	0	2	0	0
utter asked concentrat ion study program	0	0	0	0	0	0	0	0	0	2	0
utter thank you	0	0	0	0	0	0	0	0	0	0	5

Then a conversation flow of requesting a weather forecast for today. Beginning by asking directly for a specific intent about the weather forecast for today. Then the chatbot asking about a specific location as an entity, after that continuing the completion query directly to API. http://api.openweathermap.org/data/2.5/weather?q={Yogyakarta,id}&APPID={your_app_id}&units=metric[2], then translate the weather forecast description into Indonesian Language.

[1] "Fathimah API." https://api.banghasan.com/. Accessed 16 Sep. 2020.

[2] "Weather API - OpenWeatherMap." https://openweathermap.org/api. Accessed 16 Sep. 2020.



assalamu alaikum

walaikum salam saya chatbot uii, ada yang bisa saya bantu

mau tau nilai saya

boleh tau nama lengkap atau NIM nya

19917003

Analitik Big Data	3	A
Analitik Teks	3	A
Data Insight	3	A-
Deep Learning	3	A/B
Metodologi Penelitian	2	A-
Pembelajaran Mesin	3	A
Permodelan Kausal	3	A

terima kasih

terima kasih kawan

Figure 6. Chat example

5. Conclusion and Future Work

In this work, we have developed a chatbot system to help answer all queries asked by people to the university. Because of the chatbot has not been used operationally on campus, the effectiveness and the ease of use for users cannot be measured yet. So the chatbot performance is only measured through the accuracy of generating sentences that can be read correctly to reply the user. Moreover, the chatbot has a response time from the inference model which is fairly fast within less than a second. In addition, the evaluation results of the chatbot model show fairly good results near the perfect score of precision, recall and F1 which has been explained in the previous section. The scores of these numbers all reach the value of 1, it's just that in the Muslim greeting intent is 0.75 in precision score and 0.85 in f1-score, also in the intent greeting 0.85 in recall score and 0.92 in f1-score.

This research is, however, still in the first stage of development within fairly sufficient simulate data. We also have implemented deep learning models for chatbots, especially by implementing the intent classifier and entity recognition on rasa nlu also dialogue policy on rasa core, then integrating it with Facebook Messenger.

Furthermore, this chatbot needs to be developed by extending the dialogue data and more features for helping the automating task in the universities, then by directing towards the open domain chatbot, for more extensive implementation in deep learning Chatbot, to be the ‘chatbot I know everything’. Also the needs for implementing chatbot within its own platform, no limitations then and more customizable.

References

[1] Yoo S Y and Jeong O R 2020 Auto-growing knowledge graph-based intelligent chatbot using BERT *ICIC Express Lett.* vol **14** no 1

[2] Al-fakhri S *et al* 2019 Aplikasi chatbot informasi Kampus Polban menggunakan aplikasi LINE Messenger *Prosiding Industrial Research Workshop and National Seminar* vol **10** no 1

[3] Dadang W 2018 Memahami kecerdasan Buatan berupa deep learning dan machine learning *Prosiding Industrial Research Workshop and National Seminar* vol **10** no 1

[4] Santoso A and Ariyanto G 2018 Implementasi deep learning berbasis keras untuk pengenalan wajah *Emit. Jurnal Teknik Elektro* vol **18** no 1

[5] Bocklisch T, Faulkner J, Pawlowski N and Nichol A 2017 Rasa: open source language understanding and dialogue management *Preprint arXiv:1712.05181*

[6] Csaky R K 2019 Deep Learning Based Chatbot Models *Preprint arXiv:1908.08835*

[7] Prashant B P, Anil M S and Dilip K M 2017 *Online Chatting System for College Enquiry using Knowledgeable Database* undergraduate project (Maharashtra: Savitribai Phule Pune University)

[8] Thakkar J, Raut P, Doshi Y and Parekh K 2018 Erasmus AI chatbot *Int. J. Comput. Sci. Eng.* vol **6** no 10

[9] Rana M 2019 A *Chatbot Based Multi-Tier Question Answering System for Retrieving Answers from Heterogeneous Sources Using BERT* thesis (Georgia: Georgia Southern University)

[10] Jiao A 2020 An Intelligent chatbot system based on entity extraction using RASA NLU and neural network *J. Phys. Conf. Ser.* vol **1487** no 1

[11] Wu L, Fisch A, Chopra S, Adams K, Bordes A and Weston J 2018 StarSpace: embed all the things *32nd AAAI Conf. Artif. Intell. (AAAI 2018)*

[12] Lafferty J and Mccallum A 2001 Conditional random fields probabilistic models *Proc. of the 18th Int. Conf. on Machine Learning 2001 (ICML 2001)*

[13] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* vol **9** no 8

[14] Goutte C and Gaussier E 2005 A Probabilistic interpretation of precision, recall and F-Score, with implication for evaluation *27th European Conf. on IR Research (ECIR 2005)*