 ISSN: 2579-7298

Mi-Botway: a Deep Learning-based Intelligent University Enquiries Chatbot

Yurio Windiatmoko[1], Ahmad Fathan Hidayatullah[2], Dhomas Hatta Fudholi[3], Ridho Rahmadi[4]

Master of Informatics Faculty of Industrial Technology Islamic University of Indonesia Yogyakarta
[1] yurio.windiatmoko@students.uii.ac.id; [2] fathan@uii.ac.id; [3] hatta.fudholi@uii.ac.id; [4] ridho.rahmadi@uii.ac.id;

ARTICLE INFO

Article history:
Received 05 Sep 2021
Revised 30 okt 2021
Accepted 28 March 2022

Keywords:
chatbot
deep learning
chatbot framework
RNN
RASA
LSTM

ABSTRACT

Intelligent systems for universities that are powered by artificial intelligence have been developed on a large scale to help people with various tasks. The chatbot concept is nothing new in today's society, which is developing with the latest technology. Students or prospective students often need actual information, such as asking customer service about the university, especially during the current pandemic, when it is difficult to hold a personal meeting in person. Chatbots utilized functionally as lecture schedule information, student grades information, also with some additional features for Muslim prayer schedules and weather forecast information. This conversation bot was developed with a deep learning model adopted by an artificial intelligence model that replicates human intelligence with a specific training scheme. The deep learning implemented is based on RNN which has a special memory storage scheme for deep learning models, in particular in this conversation bot using GRU which is integrated into RASA chatbot framework. GRU is also known as Gated Recurrent Unit, which effectively stores a portion of the memory that is needed, but removes the part that is not necessary. This chatbot is represented by a web application platform created by React JavaScript, and has 0.99 Average Precision Score.

I. Introduction

A chatbot is the most accessible system for every user; it is an artificial intelligence technology to communicate through natural language with humans. Chatbot makes it possible to understand human language through natural language processing. Language is a protocol of social and cultural reciprocity by nature. It is time for computer machines to add this feature to human understanding along with the increasing performance of today's computer machine hardware [1].

Students or prospective students often need up-to-date information, for example asking customer service for something, especially during this pandemic, when it is difficult to have face-to-face meetings. Therefore, a chatbot is needed to answer questions quickly, correctly, and available for 24 hours [2]. Functionally, this chatbot helps in several ways such as lecture schedule info, student grade information, and several additional features for Muslim prayer schedules and weather forecast information.

The trend of chatbot development itself in its implementation for a college or campus customer service was developed by [2] using the LINE application and still using rule based mapping for chatbot response. The development of chatbots using Deep Learning for a college or campus customer service has not been found, especially in the use of Indonesian language.

To help solve the above problems, this research proposes a chatbot using a deep learning model. Deep learning consists of high-level abstract modeling algorithms on data, using non-linear function transformations that are composed of multiple layers and in depth [3]. Deep learning is also a branch of machine learning, which is inspired by the neuron of the human brain represented by layers at neural networks [4]. The deep learning method used in this research is based on RNN with a special

type called GRU (Gated Recurrent Unit), this model has several special memory saving schemes in deep learning. GRU is able to efficiently save some of the required memory and also delete some unnecessary memory. This research also uses a framework called RASA to build this chatbot application, which also integrates GRU as a deep learning method in it. The implementation of deep learning in chatbots requires the role of the chatbot framework which is to unify and simplify the process of pipelining the deep learning model and the flow of the conversation. RASA works on two main procedures namely RASA NLU and RASA Core. RASA is a library of Python Language programming that is open source for building chatbot software. The goal and philosophy of the initiator of the RASA chatbot framework is to create machine learning based dialogue management that provides ease of use in terms of implementation, as well as bootstrapping, which is to begin from existing sources to make things more complex and in a more efficient way, even starting with less data or initially with minimum training data [5].

This research is a development of [6], the model was developed through evaluation and comparison between Standard RNN, LSTM and GRU models, whereas previously only using the default of RASA framework namely RNN LSTM model. Then representation of the chatbot now uses React web apps embedded on mi-gateway UII homepage, whereas previously only using facebook messenger integration.

II. Related Works

In this section, we discuss some previous studies regarding the implementation of chatbot that have been conducted by other researchers. There are two main approaches to generate chatbot responses. If the chatbot uses a traditional approach, then it is a template rule-based for processing responses. However, there are currently many new and interesting approaches that allow deep learning to emerge. The model is a neural network, trained with some data to learn the process of generating grammatically relevant responses and in terms of understanding the user's utterances [7].

The following are some of previous works on the chat system or chatbot for universities. Online chat system for college inquiries using a knowledgeable database by [8]. In this study, they did pattern matching to search for information on chatbots. Yet using the approach of deep learning, so the rules are still too rigid for users to ask questions. However, the detailed work steps in this study are very good, there is an UML design, and various kinds of process diagrams [8]. Erasmus AI chatbot, Erasmus is a chatbot on Facebook Messenger that is used to answer questions related to college information. Designing end-to-end systems using cloud services, from api.ai (Dialogflow), Mlab (MongoDB cloud), IBM Bluemix (webhook API), scraper import.io to minimize the process coding or scripting, but what happens here, it's all because there are too many diverse cloud services, this affects quite a long latency between services cloud [9].

Eaglebot is a chatbot questioning system with Multi-Tier based, to retrieve answers from heterogeneous sources using BERT. Chatbot system that can be scalable by using 3 methods for selecting the route with the mainframe using Dialogflow, then some method of completion of document retrieval and document readers [10]. Eaglebot serves to answer various questions that are often asked by students in the university environment. However, the application Eaglebot still has some limitations by Dialogflow chatbot framework that has some limitations request, if you want to make a request unlimit basis for chatbot developers, you need a subscription for the privilege of members.

Dialogue Chatbot with entity extraction based on smart chatbot system using RASA NLU and Neural Network [11] by Anran Jiao compared the performance between the RASA NLU stack with the Neural Network Classifier and Entity Extractor which was built from scratch by the author. This study explains that the RASA NLU method is still superior for extracting all entities and classifying the user's utterances. In this study, also conducting a fairly comprehensive study, but research is only dependent on the free API in the response-completion and not using a database-completion-response.

Compared to all previous work, this research will be more reliable using deep learning for the natural language understanding of the chatbot and combining the free API with its own database for response completion. So there is no need for pattern matching, not using cloud provider-based

services, and not only relying on free API services, so that customization will be easier, scalable and there is no demand limit for user interaction.

III. Methodology

Figure 1 illustrates the diagram flow process of our study. At this stage, there is a flow and method of the research process carried out. Starting with data simulation based on common dialog interactions along with grouping each sentence into the topic intent, utter template responses chatbot, and actions. Especially in the topic intent that has been proposed as a chatbot feature, it is also necessary to define entities there. The difference between utter and action is that utter is just a sentence with an expression of a common response, whereas an action is used when the response requires database or API completion, it needs to be wrapped as a class action in RASA.

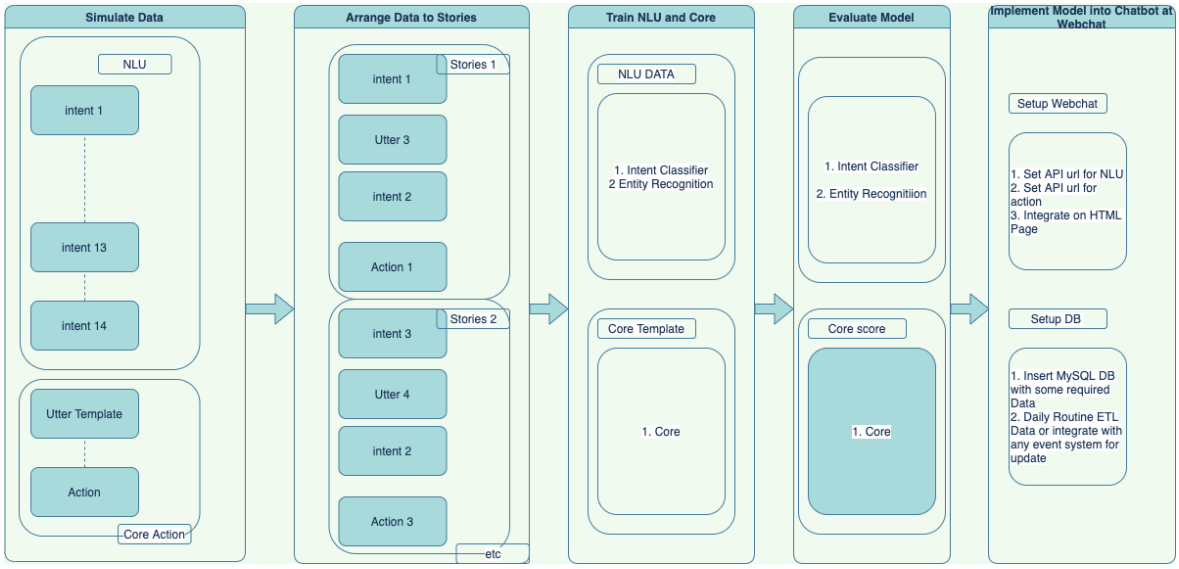

Fig. 1. Flowchart of chatbot building process [6]

Generating Data

Various data previously seen and considered from general reference schemes or conversation scenarios for each chatbot feature, such as the communication flow of student questions regarding lecture schedules, student grades, prayer schedules, and weather forecasts, are also added with several general dialogue intents such as greetings and so on. Then, the intent, utter, and action are arranged in such a way that they correspond to the storyline and general entities in each conversation sentence. an example of the data like this below.

intent request schedule:

- *Minta jadwal untuk* [Data Science] *(konsentrasi)*

- → translate: Ask schedule for [Data Science] (study_program)

- {Sentence} {Sentence} [sentence/phrase] (entity class) → nlu format RASA framework

After all the data is ready, then proceed the training using the RASA Trainer. Check and re-evaluate after the training is complete by evaluating the model being trained. If the evaluation score obtained has reached the expected expectations, it is sufficient to integrate and implement the model with the chatbot web apps platform.

Framework

Chatbot framework is used to bootstrapping all chatbot modules. Especially for the implementation of machine learning model, below is the flow architecture 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.

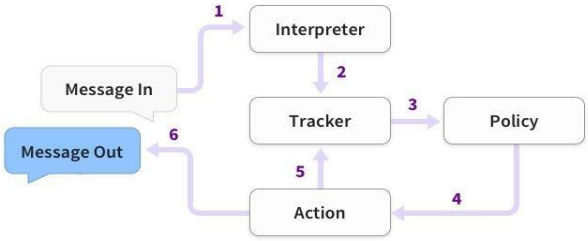

Fig. 2. Rasa stack framework

A. Modeling

Figure 3 shows the flow of training intents and entities. This chatbot system has 2 modules and each module is trained separately, these modules are NLU and Dialogue Policy or Dialogue Management. The NLU or interpreter consists of two components, namely the Intent Classifier and Entity Extractor. Process that occurs during training is started from sentence extract by tokenizer to token, then token is converted by text featurizer into vector representation to be trained on being entity extractor model by CRF entity extractor and Intent classifier model by supervised embedding.

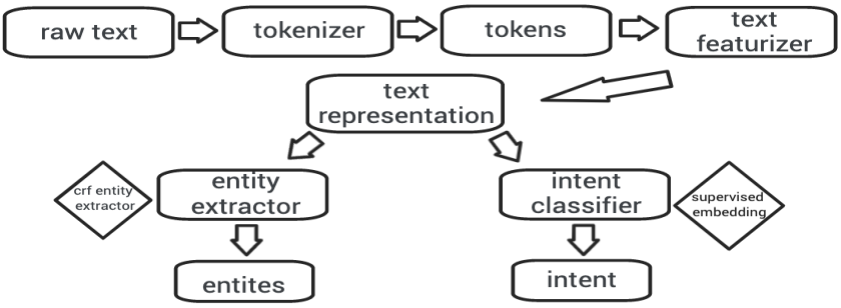

Fig. 3. Flow of training intents and entities

In particular, this chatbot uses supervised embeddings as an intent classifier, which is also known as a classifier intent embedding. The Classifier intent embedding, embeds user input and labels intent into the same dimension space. Supervised Embedding train by maximizing the similarity between the two. This algorithm is based on StarSpace [12]. However, in this implementation, the loss function is slightly different and the hidden layer is added along with the dropout. This algorithm also provides a similarity rating for the label.

The configuration of training to create this intent classifier model is set to 300 for epoch, then the hidden size layer in the neural network feed forward is 256 in h1, 128 in h2, and 14 in the intent label class on the network neural output.

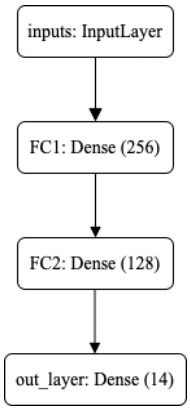

Fig. 4 NLU model for intent classification

The dimensions of embedding used for training are 20, this is multiple embedding layers in the model architecture. For example, the token vector '__CLS__' and intent are passed to the embedding layer before being compared and the loss values calculated. The configuration for the batch size is (64, 256) which are the starting and ending values for the batch size. The batch size will be increased linearly for each epoch. The learning rate is set with a value of 0.001 for configuration of training.

Entity extractor used for this chatbot is crf entity extraction [13] which is a method suitable for custom entity extractors. Its component implements the conditional random field (CRF) to perform entity recognition. The CRF can be thought of as an undirected markov chain where the time steps are words and the status is the entity class. The features are derived from words like capitalization, POS tagging, etc. They provide a probability for a particular entity class, such as the transition between tag entity and a set of tags that are most likely then to be calculated and returned. The configuration training for CRF Entity Extractor has a list of features to use. However, it can be set and selected under any conditions. Specifically, this chatbot uses a token feature list detection process such as:

- "lower case" to check if the token is lowercase.
- "Uppercase" to check if the token is uppercase.
- "Title" to check if the token starts with an uppercase character and all remaining characters are lowercase.
- "Digit" to check if the token contains only digits.
- "Prefix5" to take the first five characters of the token.
- "Prefix2" to take the first two characters of the token.
- "Suffix5" to take five last character tokens.
- "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 additional bias features to the feature list.

When the detector moves on the token in the user's message with a sliding window process, this feature defines the features for the previous, the current token , and the token next in the sliding window process. So the feature description is like [before, token, after] which is considered as an array of tokens.

As for dialogue policy or dialogue management, it works by creating data training from stories and train the models on that data. A story looks like some scenario or schematic of a common topic in any conversation the researcher designed or researcher might expect. Chat response works by creating data training from stories and training the model on that data as described in Figure 5. Dialogue policy decides which action to take at each step of the conversation. There are a variety of policies to choose from, and these include multiple policies in one agent or chatbot. At every stage, policies that predict the next action with the highest probability confidence will be used.

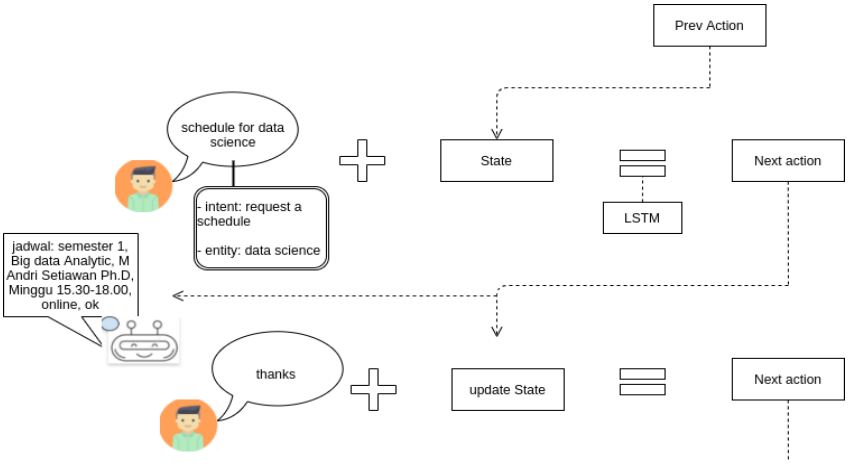

Fig. 5. Flow of dialogue management components

This chatbot is configured with a memorization policy and a neural network policy. Implementation of policy will be based on priority; the highest priority policy will be implemented first and so on. Priority is calculated on the basis of the probability confidence value of each policy, which has a higher score given higher priority. Memorization policy only remembers conversations in training data. It predicts the next action with confidence 1,0 if this exact conversation is in the training data, otherwise it predicts none with confidence 0,0. The neural network policy is then implemented to select the next response of action. The default architecture is based on LSTM [14]. This is configured with a masking layer (input, 5, 32), LSTM units (32), dropout (0.2), dense layer (19), then 100 training epoch. Then this research will compare the RNN model within the below scenario.

Table 1 Scenario for measurement RNN model

model	filter dimension
SimpleRNN	32
SimpleRNN	64
SimpleRNN	128
LSTM	32
LSTM	64
LSTM	128
GRU	32
GRU	64
GRU	128

B. Evaluation Model

Research on the development of chatbots using the deep learning model is expected to provide the best performance for the chatbot system through the deep learning model. Furthermore, the representation of the results from the deep learning model evaluation performance on the chatbot, namely the intent classification and extraction of entities at NLU also Dialogue Policy on Dialogue Systems Management of chatbot, measured and evaluated with precision value, score of recall, and score of F1 [15]. But then, because the main points of this research emphasized on the deep learning model which is implemented in the Dialogue Policy System, so in this study, there is a comparison of the performance between the RNN architecture, namely Standard RNN, LSTM, and GRU.

Confusion Matrix is a performance measurement for machine learning classification problems where the output can be two or more classes [15]. Usually the Confusion Matrix is a table with 4 different combinations of predicted values and actual values. There are four terms that represent the results of the classification process in the confusion matrix, namely True Positive, True Negative, False Positive, and False Negative. Confusion Matrix for Multi-Class Classification it's a bit different, just consider a multi-class classification problem to be a multi-class classification problem. Unlike binary classification, there are no positive or negative classes here. What we have to do here is to find TP, TN, FP and FN for each individual class. Here, it will only display data that has a misclassification value because the confusion matrix table as a whole will be less effective and consume the research paper's space paper.

Precision can be defined as the probability that an object is relevant because it is returned by the system as a predicted model, whereas recall is the probability that the relevant object is returned based on the basic truth of each label data. Then, the F1 score is needed to find a balance between precision and recall.

$$Precision = \frac{True\ Class}{True\ Class + False\ Predicted\ Class} = \frac{True\ Class}{Total\ Predicted\ Class} \quad (1)$$

$$Recall = \frac{True\ Class}{True\ Class + False\ Predicted\ Other\ Class} = \frac{True\ Class}{Total\ Actual\ Class} \quad (2)$$

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

In short, that precision shows how precise or accurate the model is outside the predicted label class, how many of which are actual label classes. Meanwhile, recall calculates how the model captures many of the actual classes by labeling it as that class. Finally, the F1 score was used to find a balance between precision and recall.

C. Platform Module Connector

Figure 6 is an illustration of web apps as a platform connector module. Settings webhook messaging platform and postback messaging. To create a callback, enter a URL that looks like https://<HOST>/webhooks/webhooks. Enter and verify webhook, it must match the verification entry in the credentials.yml file. So, webhook allows chatbots to receive real-time HTTP notifications about changes to specific objects and send them back to messaging on the web platform.

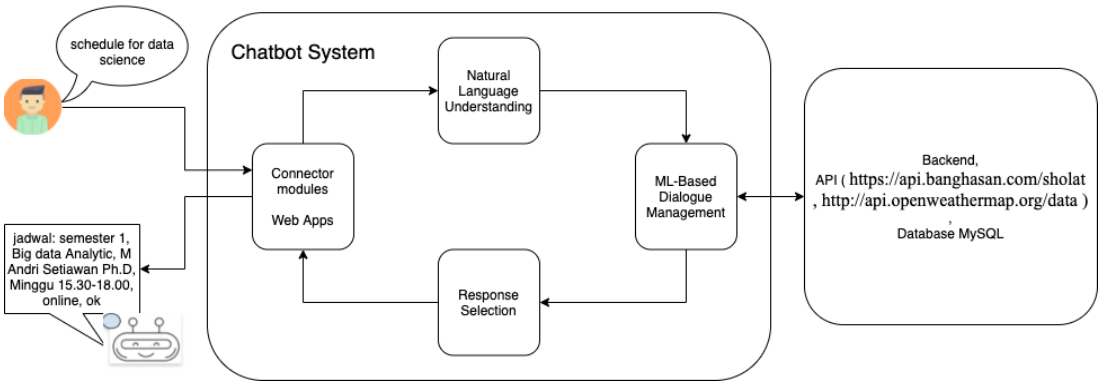

Fig. 6. Flow of chat system

IV. Result and Discussion

The table below is the result of evaluating chatbot metrics on intent classifier , entity recognition and chatbot response models that compare the RNN models.

Table 2. Evaluation metric of Intent Classifier 1

metrics	intent goodbye	intent confirm city	intent confirm name or nim	intent worship schedule	intent refuse	intent greeting	challenge bot	good mood
precision	1	1	1	1	1	0.88	1	1
recall	1	1	1	1	1	1	1	1
f1-score	1	1	1	1	1	0.93	1	1

Table 2 shows the metric intent evaluation for the classification process. It is necessary to pay attention to the class intent Muslim greeting where the model can accurately predict the amount of truth data for that class with a recall value of 1, but the model still predicts the Muslim greeting intent in other actual intent classes with an accuracy score of 0.75, namely intent greeting class.

Table 3. Evaluation metric of Intent Classifier 2

metrics	intent request grade score	intent request schedule	intent request schedule only	intent weather prediction	intent muslim greeting	intent agree	bad mood	thank you
precision	1	1	1	1	1	1	1	1
recall	1	1	1	1	0.67	1	1	1
f1-score	1	1	1	1	0.8	1	1	1

In Table 3, the evaluation metrics for describing the intent classification process. As for the intent greeting itself, it gets a precision score of 1 where the intent greeting sometimes not detected by the model, otherwise the intent Muslim greeting detected because it uses the same word phrase, namely 'assalamu alaikum' but this is not a problem because of the template utter salam chatbot will answer the same thing as the utter greeting.

Table 4. Confusion Matrix between misclassified label classes

Predict VS Actual	Intent Muslim greeting	Intent greeting	Data Count	Recall
Intent Muslim greeting	3	0	3	1
Intent greeting	1	6	7	0.85
Total Prediction	4	6	-	-
Precision	0.75	1	-	-

Table 4 provides a more detailed explanation of the precision and calculations recall of the two frequently misclassified label classes. It can be seen from table 3 that the precision is equal to 1. This shows that our chatbot system is able to detect the intent greeting correctly. As for the intent Muslim greetings, the score precision is 0.75.

Table 5. Evaluation metric of Entity Recognition

	Name	Study Program	City	NIM	Average	Total
Precision	1	1	1	1	1	4
Recall	1	1	1	1	1	4
F1-Score	1	1	1	1	1	4

in table 5, evaluation metric of entity class has shown the best performance of the model. It already has the ability to extracting each of entity labels from each sentence.

Conversation flow process is divided into various flows or stories, based on the capability or features of the chatbot. First the schedule feature, students who request a lecture schedule from the chatbot have the following process. The chat starts with student makes greeting to the chatbot, and the chatbot detects the intent from the user and replies with a reply utter greeting. Then students get to the point to ask about class schedules, and the chatbot detects it as an intent to request scheduling from the user, then answers with utter asking about the concentration of the student study program. Then the students tell the chatbot about the concentration of their study program, namely 'fd' or digital forensics, and the chatbot considers this as the entity study program concentration to continuing the entity through query MySQL DB as a completion to get schedule data.

The process of requesting a schedule can also be passed by directly asking for the concentration schedule of a particular study program, for example 'ds schedule' and the chatbot will detect the intent to request the schedule as well as detect the entity concentration of the study program data science. Then the chatbot continuing through query on MySQL DB as a completion to get lecture schedule data.

For example, conversations of students asking about grades score. This chat flow is about requesting a student grade list, it is almost the same as the flow requesting a class schedule, but only through different intent and entity, as well as a different action to display a list of grades,

entity name or NIM is continued to the completion query through MySQL DB as a complement to get a list of the corresponding student scores.

In addition, there are also conversations to ask for prayer times. Starting with asking directly with the intention of asking the Muslim prayer schedule. Then the chatbot asks for a specific location as an entity, after which it continues the settlement query directly to the API https://api.banghasan.com/sholat/format/json/jadwal/kota/[city]/tanggal/[date.now].

Then the conversation flow asks for today's weather forecast. Starting by asking directly with an intent about today's weather forecast. Then the chatbot inquires about a specific location as an entity, after which it proceeds to direct the completion query to the API. http://api.openweathermap.org/data/2.5/weather?q=[Yogyakarta,id]&APPID=[your_app_id]&units=metric, then the results of return api are translated into Indonesian.

Table 6. Comparison of RNN Architectures performance

model	filter dimension	ap	ar	af	avg_all
GRU	128	0.9925	0.9793	0.9831	0.9888
SimpleRNN	32	0.9843	0.9912	0.9862	0.9875
SimpleRNN	128	0.9843	0.9912	0.9862	0.9874
LSTM	32	0.9843	0.9912	0.9862	0.9873
GRU	32	0.9843	0.9912	0.9862	0.9872
LSTM	64	0.9843	0.9912	0.9862	0.9872
LSTM	128	0.9843	0.9912	0.9862	0.9862
GRU	64	0.9925	0.9793	0.9831	0.9855
SimpleRNN	64	0.9925	0.9793	0.9831	0.9844

various types of RNN architectures. Comparison is carried out against the architecture Standard RNN, LSTM, and GRU by considering the value of RNN size at 32, 64 and 128 in each architecture. Here, it shows that the evaluation of the entire architecture is not that far away, but the best performance value is achieved by the GRU 128 RNN size with a score of Average Precision 0.9925, Average Recall 0.9793, Average F1 0.9831.

Table 7. Comparison GRU-128 and LSTM-32 default framework

GRU-128	LSTM-32	GRU-128	LSTM-32	GRU-128	LSTM-32
college student:	college student:	*mi-botway:*	**mi-botway:**	**(translate: score list**	**(translate: thanks accepted :))**
cek nilai	cek nilai	*daftar nilai*	**kasih diterima :)**	**Big Data Analytics course 3 A**	
(translate: grade score check)	(translate: grade score check)	*mata kuliah Analitik Big Data 3 A*		**Text Analytics course 3 A**	
mi-botway:	mi-botway:	*mata kuliah Analitik Teks 3 A*		**Data Insight course 3 A-**	
minta nama lengkap atau NIM nya dong :)	tuliskan nama lengkap atau NIM nya ?	*mata kuliah Data Insight 3 A-*		**Deep Learning course 3 A .**	
(translate: ask for full name or NIM please :))	(translate: write full name or NIM?)	*mata kuliah Deep Learning 3 A*		**Research Methodology course 2 B+**	
college student:	college student:	*mata kuliah Metodologi Penelitian 2 B+*		**Machine Learning course 3 A**	
yurio windiatmoko	yurio windiatmoko	*mata kuliah Pembelajaran Mesin 3 A*		**Causal Modeling course 3 A)**	
(translate: yurio windiatmoko)	(translate: yurio windiatmoko)	*mata kuliah Pemodelan Kausal 3 A*			

Table 7 shows the comparison of chat responses between GRU-128 and LSTM-32 as the default framework model, GRU-128 model also provides a proper response compared to the default LSTM-32 model according to previous research [6]. This implementation is actually on web chat at mi-gateway homepage UII, so that this chatbot is called mi-botway. The position of the chatbot module is on the html body of mi-gateway homepage at the bottom.

V. Conclusion and Future Work

In this research, a chatbot system has been developed to help answer some of the questions raised by various parties to the university. Because of the chatbot has not been used operationally on campus, the effectiveness and ease of use for users cannot be measured. So that, chatbot performance is only measured by its accuracy in producing sentences that users can read correctly and precisely. In addition, the chatbot has a fairly fast response time from the inference model in less than one second. In addition, the evaluation results of the chatbot model show fairly good results close to the perfect score for precision, recall and f1 which have been described in the previous section. The scores of these numbers all reach a value of 1, it's just that the intent Muslim greeting is 0.75 on the precision score and 0.85 on the f1 score, also on the greeting intent 0.85 on the recall score and 0.92 on the f1 score.

However, this research is still in the first development stage with sufficient simulation data. This research has also implemented a deep learning model for chatbot, especially by implementing the intent classifier and entity recognition on the RASA NLU and the dialogue policy on the RASA core, then integrating it with the web apps platform.

For the future work, we suggest to develop by extending dialogue data and more features to assist automation tasks in universities, then by evolving and directing the chatbot as an open question domain, for wider application to become 'chatbot i know everything'.

References

[1] S. Y. Yoo and O. R. Jeong, "Auto-growing knowledge graph-based intelligent chatbot using BERT," *ICIC Express Lett.*, vol. 14, no. 1, pp. 67–73, 2020, doi: 10.24507/icicel.14.01.67.

[2] S. Al-fakhri *et al.*, "Aplikasi Chatbot Informasi Kampus Polban Menggunakan Aplikasi LINE Messenger," no. November, pp. 302–313, 2019.

[3] W. Dadang, "Deep Learning (Pembelajaran Dalam)," *06 februari 2018*, 2018. https://warstek.com/2018/02/06/deepmachinelearning/ (accessed Jun. 06, 2020).

[4] A. Santoso and G. Ariyanto, "Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah," *Emit. J. Tek. Elektro*, vol. 18, no. 01, pp. 15–21, 2018, doi: 10.23917/emitor.v18i01.6235.

[5] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," pp. 1–9, 2017, [Online]. Available: http://arxiv.org/abs/1712.05181.

[6] Y. Windiatmoko, R. Rahmadi, and A. F. Hidayatullah, "Developing Facebook Chatbot Based on Deep Learning Using RASA Framework for University Enquiries," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012060, 2021, doi: 10.1088/1757-899x/1077/1/012060.

[7] R. K. Csaky, "Deep Learning Based Chatbot Models," 2019, [Online]. Available: http://arxiv.org/abs/1908.08835.

[8] B. P. Prashant, M. S. Anil, and K. M. Dilip, "Online Chatting System for College Enquiry using Knowledgeable Database PARTIAL FULFILLMENT OF THE REQUIREMENTS OF BACHELOR OF ENGINEERING (Computer Shri Chhatrapati Shivajiraje College of."

[9] J. Thakkar, P. Raut, Y. Doshi, and K. Parekh, "Erasmus AI Chatbot," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 10, pp. 498–502, 2018, doi: 10.26438/ijcse/v6i10.498502.

[10] M. Rana, "Digital Commons @ Georgia Southern EagleBot : A Chatbot Based Multi-Tier Question Answering System for Retrieving Answers From Heterogeneous Sources Using BERT," pp. 1–54, 2019, [Online]. Available: https://digitalcommons.georgiasouthern.edu/etd.

[11] A. Jiao, "An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network," *J. Phys. Conf. Ser.*, vol. 1487, no. 1, 2020, doi: 10.1088/1742-6596/1487/1/012014.

[12] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "StarSpace: Embed all the things!," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, pp. 5569–5577, 2018.

[13] J. Lafferty and A. Mccallum, "Conditional Random Fields Probabilistic Models," vol. 2001, no. June, pp. 282–289, 2001.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] C. Goutte and E. Gaussier, "Ch10_Witnesses[8463].Pdf," no. April, 2005, doi: 10.1007/978-3-540-31865-1.