

# Prototyping a Chatbot for Student Supervision in a Pre-Registration Process

Lucia Dwi Krisnawati<sup>1</sup>, Bill Edward Butar-Butar<sup>2</sup>, and Gloria Virginia<sup>3</sup>  
<sup>1–3</sup>Information Technology Department, Duta Wacana Christian University  
Yogyakarta 55224, Indonesia

Email: <sup>1</sup>[krisna@staff.ukdw.ac.id](mailto:krisna@staff.ukdw.ac.id), <sup>2</sup>[billyedward15@gmail.com](mailto:billyedward15@gmail.com),  
<sup>3</sup>[virginia@staff.ukdw.ac.id](mailto:virginia@staff.ukdw.ac.id)

**Abstract**—Developing a chatbot becomes a challenging task when it is built from scratch and independent of any Software as a Service (SaaS). Inspired by the idea of freeing lecturers from the burden of answering the same questions repetitively during the pre-registration process, this research has succeeded in building a text-based chatbot system. Further, this research has proved that the combination of keyword spotting technique for the Language Understanding component, Finite-State Transducer (FST) for the Dialogue Management, rule-based keyword matching for language generation, and the system-in-the-loop paradigm for system validation can produce an efficient chatbot. The chatbot efficiency is high enough as its score on Concept Efficiency (CE) reaches 0.946. It shows that users do not need to repeat their utterances several times to be understood. The chatbot performance on recognizing new concepts introduced by users is also more than satisfactory which is presented by its Query Density (QD) score of 0.80.

**Index Terms**—Chatbots, Dialogue System, Keyword Spotting Technique, Transducer

## I. INTRODUCTION

CHATBOTS, also known as conversational agents or chatterbots, are computer applications that imitate human personality [1]. Chatbots also enable online human-computer dialog with natural language [2]. Recently, chatbots have become popular and attracted the interest of many researchers, companies, and users. This is proven by the fact that by September 2016, Facebook messenger had hosted 30.000 bots and had 34.000 developers on its platform [3]. Meanwhile, Jemma, a chatbot released by [kata.ai](http://kata.ai) for Unilever Company, has sent 50 million messages and got 17 million friends in less than one year period [4].

At least three factors trigger the rise of chatbots. First, their ability to interact intelligently with humans has improved significantly [5]. Second, it is the advancement of hardware technologies and artificial intelligence supports. Third, it characterizes the era

in which there are plenty of accessible open source codes. The development platforms are also available widely, and chatbots implementation options are available through Software as a Service (SaaS) [6] such as Amazon Lex. These factors make chatbots are now easier to train and implement.

Although chatbots gain popularity in recent years, their existence can be traced back to 1966 when Weizenbaum introduced ELIZA which was programmed to act as a Rogerian therapist. ELIZA was able to fool users into believing that they were conversing with a real human being [7]. Another notable chatbot is Artificial Linguistic Internet Computer Entity (A.L.I.C.E) which was written in 1995 using AIML (an XML-based markup language). The modern chatbots have a wide range of functions, the degree of intelligence and modalities, whether they are text-based or voice-based conversational agents.

Both text and voice-based chatbots are one category of conversational agents which are not embodied in the forms of animal, avatars, or human robots [6]. The one which devises computer-generated cartoonlike characters is categorized as embodied conversational agents [8]. The conversational agents fall into a class of Dialogue System (DS) which has been subject to research for decades. There is another class of DS which is not categorized as conversational agents such as Interactive Voice Response (IVR). The exclusion of IVR from the conversational agents is caused by its modality of interaction which uses phone keypads (“press 1 to choose xxx”) instead of utterances. Reference [8] also constructed the taxonomy of DS which can be seen in Fig. 1.

As a class of DS, chatbots simulate a conversation in its primary sense, intend to fool users with whom they are communicating [9]. In response to robustness, pattern matching techniques are commonly used to provide a certain level of control over system [2, 9]. In Contrast, DSs attempt to model the actual dialogue process which incorporates the task of analyzing

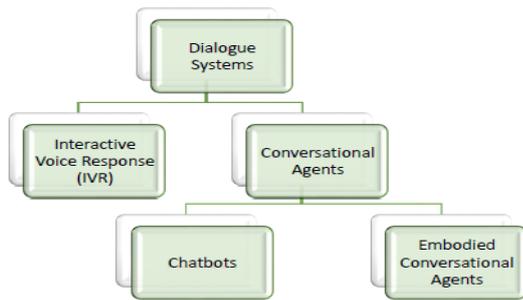


Fig. 1. The class hierarchy of dialogue systems [8].

and understanding input. DSs make use of refined technologies and approaches including the integration of knowledge, ontologies, and the use of methods originating from Computational Linguistics [9].

This research focuses on implementing a text-based chatbot for student supervision in a pre-registration process. The goal is to free the academic supervisors from the burden of answering the same repetitive questions from different students. The chatbot devises a Keyword-Spotting technique to understand the user inputs and Finite-State Transducer (FST) for managing the dialogue run. To gain mutual understanding between users and our chatbot, its Dialogue Manager (DM)r component is completed with event handling and verification process. It is built from scratch and can be run from the local server. Thus, the chatbot is independent of pay-per-user as a SaaS pricing strategy or other software license billing.

## II. LITERATURE REVIEW

A literary survey on chatbot by Ref. [10] has concluded that the techniques of chatbot designs are still a matter of debate due to its varied approaches. However, it is inferred that the basic architecture of a chatbot follows its parent, DS. It comprises three main components. Those are a natural language understanding (NLU), DM, and a natural language generation (NLG) [5]. NLU processes raw user inputs and extracts information into a semantic representation that can be interpreted by DM [11] to update the internal states, send queries to the database, or find actions based on scripts [5]. Some methods and approaches commonly applied to NLU are semantic formats which represent an utterance in key-value pairs [1, 12], or a template matching between user input and pre-defined utterances [13]. A more flexible matching approach takes a form of keyword matching [14], or Data-Driven Approach which requires a large corpus of interactions and utterances as found by Ref. [5].

The primary task of DM is to interpret semantic representation outputted by NLU in the context of

dialogue to decide the actions. The various methods applied in building DM can be categorized into three approaches: the finite-state, frame-based, and statistical approaches [11]. The finite-state approaches tend to lead a deterministic dialogue flow and have a single-initiative dialogue, in which system or user takes the dialogue control. The implementation of AIML-based chatbots as in [12, 13, 15] are mostly deterministic and single-initiative. Hence, they are claimed to fall into the category of finite-state approaches.

The frame-based system works with a frame consisting of slots. The dialogue flow is not pre-defined as in finite-state system. Thus, it enables users to exchange initiative or control over the conversation with the system or a mixed-initiative dialogue [8]. Based on frame-based DM, Krisnawati in [14] successfully elaborated the capabilities of her mixed-initiative DS into performing a real subtask action such as dialing the extension number of certain staff demanded by the user in the dialogue. Meanwhile, Ref. [16] combined the mixed-initiative dialogue with a knowledge-based DM which kept track of the current state of the conversation.

The NLG acts inversely to NLU. It is responsible for presenting responses generated by the DM to users. In the latest systems applying statistical and machine learning approach, the tasks of NLU, DM, and NLG are performed by a single function. For example, Ref. [5] devised a single deep network to merge the task of three components. Meanwhile, Ref. [17] applied Maximum Entropy and Gibbs Distribution to represent and select the user-system sequence pair of dialogues. The IBM researchers propose a futuristic DS using a new dialog programming model based on grammars [18]. They argued that grammar, which was a successful formalism of imposing a structure over sequences of conversation, could solve the human-computer dialogue problems.

So far, text or voice-based chatbots function as a Question-Answering agents whose task is to retrieve the information needed by users and end the dialogue after the answer is delivered. This can be found in Apple’s Siri, Google Now apps, or in [15]. In contrast, the Information Retrieval-based DS has more complicated dialogues with users such as in [1] which delivered queries on book search and library services, or in [11] which reserved a movie ticket. A chit-chat with no specific topic for achieving fun and user entertainment can be done by both chatbots and DS as found in Ref. [13]. Other functions taken over by DS is a healthcare coach and advisor for retired troops [16], customer service of a company such as Vodafone [5], technical assistant and troubleshooter in using computer devices [17], and online shop-assistant [19].

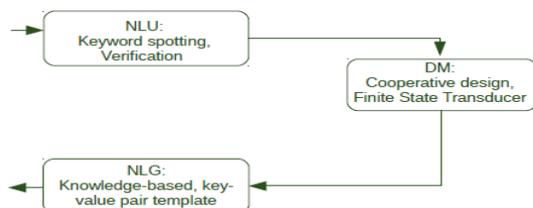


Fig. 2. The basic architecture of the chatbot.

### III. RESEARCH METHOD

In the absence of a conversation corpus, implementing a chatbot with a domain-specific dialogue is much more controllable than a chit-chat one. Based on this argument, the researchers construct a text-based chatbot. It focuses on coping with two topics of pre-registration process such as the maximum credits for a student who is eligible to enroll, and the subjects offered. These topics are chosen to comply with the research objective. It is to relieve the supervisors from answering the same repetitive questions. Moreover, those topics fall under the most frequently asked questions among students.

Another research objective is to construct a chatbot that is free from SaaS. For this reason, the researchers apply a set of simple but applicable methods in each module of chatbot implementation and provide possibilities to upgrade their capabilities. The basic architecture of the chatbot in Fig. 2 follows the general architecture of DS.

#### A. Data Collection

One of the problems in developing a chatbot system is how to model the human-machine dialogue flow. This implies that modeling a dialogue flow of a chatbot needs data taking the form of conversations. To collect such data, the researchers need a system prototype in which users can communicate. This becomes a chicken and egg problem.

To break the cycle of this chicken-egg problem, the researchers firstly observe how students usually pose questions to their academic supervisor. Then, the question-answer formulations are sampled from a handful of IT students by asking them to playact the conversation. The results are some topics of pre-registration questions. The researchers pick up two categories which are most frequently asked. Those are the number of credits 'Satuan Kredit Semester' (SKS) and the offered subjects. Based on this restricted data, the researchers design a chatbot with insufficient natural language capability and dialogue flows. This chatbot is used to collect data through "system-in-the-loop" paradigm.

System-in-the-loop, which was introduced by Ref. [20], is a wizardless and iterative procedure for collecting data using the developing system prototype. The researchers apply this data collection paradigm in two iterations, each with ten different users. Most users are students, but there are two lecturers involved in this process. For data gathering, a task scenario consisting of guidance on doing the given tasks is prepared. Besides, Camtasia (a screen recording software) is installed to record and capture any user movement on the screen such as what they type, and how they converse with the chatbot. After interaction with the pre-alpha chatbot, users are interviewed to give feedback on the interface design, dialogue navigation, and the system capability in dealing with the dialogue.

Nine out of ten users in the first session suggest dialogue navigation that enables moving backward to the nodes leading to the former topic of conversation. The resulting data are used to improve the system navigation which is completed with verification. Moreover, the unrecognized user inputs in this session are used for enriching the vocabulary of the natural language understanding component as synonyms for the defined keywords.

Using the same procedure but improved task scenario, the refined chatbot is run for the second test for ten different students from the first test. In this session, the users' feedbacks on system improvement become more specific and focused. For example, the backward navigation in a node of compulsory subjects needs improvement. It is due to its being error-prone. Moreover, the system should also understand the writing variations of keywords, the abbreviations of subject names, and the use of Arabic numbers as an alternative for stating numbers. Most feedbacks in this session are from both interview and user interaction with the chatbot and deal with the improvement on NLU and DM component. Only three users suggest the improvement of the interface design.

#### B. Keyword Spotting Technique

Processing and understanding user input are the tasks of language understanding component of a chatbot. Most currently built chatbots accept user inputs in the form of sentences instead of phrases and word sequences. There is much variety of methods to understand these sentences. Some systems treat the whole user input sentence as a template to be matched. In Ref. [13], a pair of user-system utterances are predefined in AIML format and saved in a database. Thus, to understand user input of "Apa kabar?", the system computes the bigram similarity of this string to all defined templates in the database (*apa kabar, siapa*

TABLE I  
THE EXAMPLES OF RECOGNIZED USER INPUT VARIATIONS IN THE CHATBOT'S LOG.

Variation of users' utterances for one query	Recognized keywords
<i>Berapakah SKS yang bisa saya ambil?</i> (how many credits that I can take?)	SKS, ambil (credit, take)
<i>Berapakah syarat pengambilan SKS maksimal tiap semester?</i> (What is the maximum taken credit of each semester?)	ambil, SKS (take, credit)
<i>berapa batas pengambilan SKS di semester ini?</i> (what is the limit of taking credit in this semester?)	ambil, SKS (take, credit)
<i>Jumlah SKS untuk IPK 2.9</i> (Number of credits for GPA 2.9)	SKS (credit)
<i>Tolong tampilkan syarat pengambilan SKS</i> (Please show credit taking requirements)	ambil, SKS (take, credit)

nama kamu, and others), and retrieves the answer of the template whose similarity score is the highest.

Instead of treating the whole user input as a template to match, this research resorts to keyword and phrase spotting method. In this method, the system needs to identify the keywords and pattern match those keywords against a set of pre-programmed rules to generate the appropriate responses. Thus, NLU component does not need to analyze an utterance fully. The advantages of this technique are that the chatbot system recognizes all utterance variations as long as they contain the keywords, and users get a positive impression on the system intelligence. The order of keyword occurrences is also ignored. Table 1 shows the variations of recognized user inputs by the chatbot.

In its implementation, at least one keyword is defined for each step of dialogue. The variation of these keywords are saved in an array and formulated as a pattern using Regular Expression to match. In total, there are 23 sets of keywords with a minimal set cardinality of 2 keyword variations and maximal cardinality of 12 variations for the keyword 'terima kasih' (thank you). Keywords taking the form of phrases are treated as separate tokens and defined only in their root word forms. As its consequence, the order of keywords in their occurrences and affixation will not affect the recognition.

### C. Dialogue Strategies

The dialogue flow of the chatbot is managed by an unweighted Finite State Transducer (FST) which is a variation of a Finite State Automata (FSA). It is capable of producing outputs and reading inputs as well. In contrast, FSA is only capable of recognizing for matching patterns. The state traversal within FST can be deterministic as well as non-deterministic depending on the applied algorithm.

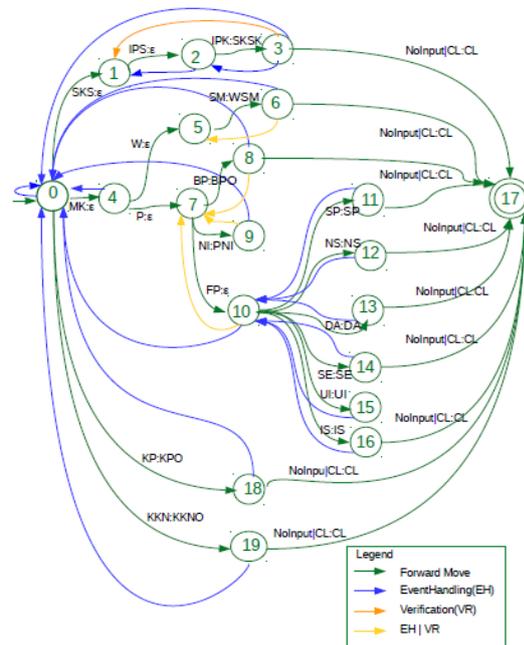


Fig. 3. The transducer's state transitions which depict the dialogue flow of the chatbot.

In the FST-based chatbot, user input is placed in one state at a given time. The chatbot maintains the control of dialogue by producing prompts at each state, and the user needs to give responses to move to another state. The recognized keywords determine the state that will be traversed in the users' responses. The transducer describing the flow of dialogue in the chatbot is seen in Fig. 3.

All states in Fig. 3 are labeled in Arabic numbers. The state labeled 0 symbolizes the start state. Meanwhile, 17 is the end or stop state. The start state has four forward transitions (green arrows) to states 1, 4, 18, and 19. The first two states deal with the main topics of conversation. Meanwhile, states 18 and 19 are the short-cut states as a result of cooperative design by integrating users' needs. The start state (0) conveys a discourse opening which users are welcome, and the domain of conversation is introduced. Figure 4 shows the capture of chatbot's prompt to start state.

The transition from state 0 to state 1, 2, and 3 marks the dialogue on how many SKS a student can take based on their semester Grade Point Average (GPA) 'Indeks Prestasi Semester' (IPS) (state 2) and cumulative GPA 'Indeks Prestasi Kumulatif' (IPK) (state 3). The state transitions from 0 to 416 regulate the Question-Answering (QA) dialogues on the offered subjects per semester, the subject types (optional or compulsory), and profile-based subject categories. The shortcut dialogue on the prerequisite of taking Intern-

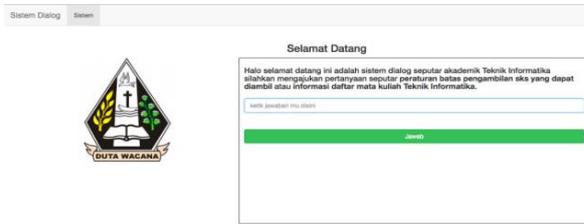


Fig. 4. The discourse opening introduced by the chatbot. A user can type her/his response on the blank text field to communicate with the chatbot. The figure in English: Hello and welcome, this is the dialogue system of Information Technology academic, please submit the questions regarding the rule of registering SKS which you can take or information about list of course in Information Technology.

ship 'Kerja Praktek' (KP) as the subject is directed for state 0 to state 18. Meanwhile, the transition to state 19 from state 0 rules the shortcut QA dialogue on the Community Service subject 'Kuliah Kerja Nyata' (KKN).

The transition from one state to another one is determined by two parameters such as the recognized keywords and the rules predefined on each state. Algorithm 1 exemplifies the transition rules on state 3. Given the inputs of the values of the semester and cumulative GPAs (IPS and IPK), state 3 prompts a verification. The user response to system verification determines the transition to other states. Algorithm 1 is presented in Alg. 1. State 3 also deals with verification and event handler which are provided on different conditions.

---

**Algorithm 1:** Transition rules of the state 3.

---

```

Input: IPS, IPK, userResponse
Output: Transition from  $X$  to  $Y$ , retrieved answer of the query
Prompt verification defined for state 3
VerifyAnswer  $\leftarrow$  read(userResponse)
if BenarKey is in VerifyAnswer then
    AnswerCand  $\leftarrow$  retrieveMatrix(IPK, IPS)
    Answer  $\leftarrow$  join(AnswerCand, templateAnswer)
    display Answer to users
    prompt user whether to end or repeat the dialogue
    if User ends dialogue OR NoInput  $\geq$  20 sec then
        random(closing statements)
        move to state 17
    else
        updateState(3,0)
        move to state 0
    end
else if SalahKey is in VerifyAnswer then
    unset(IPK, IPS)
    updateState(3,1)
    Move to state 1
else if No IPKkey found OR IPKformat is false then
    EHPrompt  $\leftarrow$  random(eventHandler alternatives)
    display EHPrompt
    unset(IPK)
    updateState(3,2)
    Move to state 2
end

```

---

The researchers use MySQL as a database to store the information on the states being traversed, and the user inputs on IPS and IPK values. The current active

state is dynamically updated as the dialogue between chatbot, and a user is in progress. The values of IPK and IPS are stored in the query regarding the total number of credits that a student (user) can take in the IPK-IPS matrix. This matrix construction is based on the academic handbook given to first-year students. Information on that handbook also defines the subject categorization and requirements.

#### D. Grounding and Verification

Grounding, which is a way of establishing mutual knowledge, is a vital part of communication in both human-human and human-computer conversation. Grounding becomes a real challenging task of having a chat with a bot, as it involves efforts to share each speaker's common understanding to achieve the goal. Reference [21] proposed three kinds of grounding strategies: the cautious grounding, the optimistic grounding strategy, and the verification. Each strategy had its strengths and drawbacks. The cautious grounding strategy would be annoying since system always checked the correctness of each user input and asked the user to confirm it. In contrast, the optimistic grounding strategy could lead to a dead end dialogue, since the system was too optimistic about the correctness of user input without confirmation.

In this chatbot, the verification is applied to ensure that the users' responses understood by the system are correct, so that the system can retrieve the right information needed by users. The verification is placed at states 3, 6, 8, 9, and 10 where users have completed all queries needed by the system to output the information as an answer to those queries. For example, to retrieve the maximal number of credits (SKS) that a student is eligible to take, the system needs the student IPK and IPS. Having received the values of IPK and IPS from the former states, the system issues the verification on state 3. This requires user confirmation by typing 'yes' or 'no'. Figure 5 captures the verification process implemented in chatbot verification.

#### E. Event Handling

Another communication strategy applied in the chatbot is the mechanism to handle events which are not covered by the state mechanism. The researchers define three events to be handled. Those are no keyword, no input, and no well-formed input.

- 1) No-keyword. When the system finds no keyword on user responses, a special prompt defined for no-keyword event will be issued. Following it, the DM directs the conversation back to the former state so that the user can correct her/his responses. There are three statements defined in

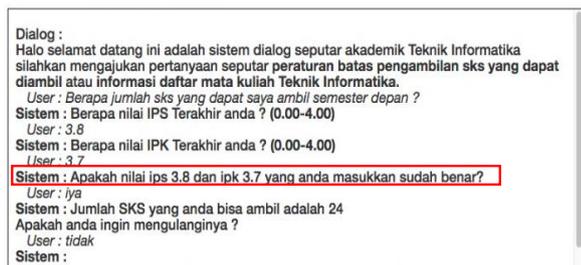


Fig. 5. the verification implementation on the system utterance inside the red box by stating “Apakah nilai IPS 3.8 dan nilai IPK 3.7...” (Is the semester GPA 2.8 and Cumulative GPA 3.7 ...). The figure in English: Dialogue: Hello and welcome, this is the dialogue system of Information Technology academic, please submit the questions regarding the rule of registering SKS which you can take or information about list of course in Information Technology. **User:** how many SKS that I can take in the next semester? **System:** what is the latest of your IPS? (0.00–4.00) **User:** 3.7 **System:** what is the latest of your IPK? (0.00–4.00) **User:** 3.8 **System:** are your IPS 3.8 and IPK 3.7 correct? **User:** yes **System:** the amount of SKS that you can take is 24 do you want to repeat the process? **User:** no **System.**

response to no-keyword events. The issuance of this statement is done through a random selection technique. The aim is to provide various event handling statements in the case in which a user makes mistakes repetitively. The following is the translation version of statements for no-keyword events:

- “I am really sorry for being unable to understand your response. Could you please check your input again?”
  - “The response you have just inputted could not be understood by the system.”
  - “Would you like to rewrite your response in an understandable way?”
- 2) No-input. In case, when a user cannot continue his/her dialogue because of many reasons that it leaves the users’ text field unfilled or blank, a no-input event will be issued. The toleration for the no-input event is set up for 20 seconds. After 20 seconds, there is no input. The DM will lead the transition to the end state 17. The no-input event function is called on every state except the start (state 0) and the state 17. Thus, there is a transition from every state to state 17, but these transitions are not depicted in Fig. 3 to avoid the crowdedness of arrows as a transition symbol. There is only one statement defined in response to no input-event such as “Your time is up, and thank you for chatting with me.”
- 3) Not well-formed input. This is to tackle an event in which the keywords are successfully extracted from the users response, but they are not well-

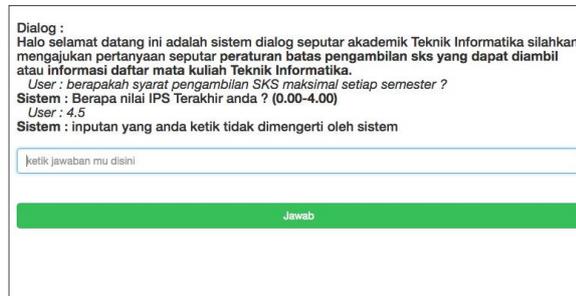


Fig. 6. An example of issuing an event handler on not well-formed input. The figure in English: Dialogue: Hello and welcome, this is the dialogue system of Information Technology academic, please submit the questions regarding the rule of registering SKS which you can take or information about list of course in Information Technology. **User:** what is the maximum SKS in each semester? **System:** what is the latest of your IPS? (0.00–4.00) **User:** 4.5 **System:** your input cannot be understood by the system.

formed. An excellent example of the not well-formed input is a conversation occurring in states 1, 2, and 3. In these states, a user inputs his/her IPS which is set in numeric format between zero (0.00) to four (4.00) since there will be no IPS greater than 4.00. If a user inputs 4.5 or -1.00 for his/her IPS, the function of not well-formed input event will be issued, and the prompts to correct the input will be done. Figure 6 illustrates a dialogue with this event handler. The followings are some translated examples of system prompts for handling not well-formed inputs:

- “Your IPS is not well-formed. Please input the right one.”
- “So, have you inputted the right IPK of yours?”
- “The input that you have provided could not be understood.”

## IV. RESULTS AND DISCUSSION

### A. Evaluation Process

The evaluation process of this chatbot system takes the form of process validation and system assessment. The process validation is applied to collect data with a goal to improve chatbot performance in having a dialogue with users. To elucidate it, this validation is considered as an iterative evaluation which is a part of system development. It has been done in two phases with ten different testers involved in each phase. All testers are IT students of different intake years. This has been done on purpose for two reasons. Firstly, IT students have the better sense of the bugs and system performance compared to students from different departments. Secondly, the researchers badly need a

TABLE II  
EXAMPLES OF TRANSLATED RESPONDENTS' FEEDBACKS FOR SYSTEM IMPROVEMENT.

Interface design	Dialogue navigation	Dialogue competency
The UI design is too simple	I cannot repeat taking the compulsory subjects	The system prompts for options should be stressed (bold, maybe) to make users easily understand
If it is possible, do not display the history of chatting	Please add navigation to repeat the dialogue to the start after finishing it	The system has to understand writing variations such as lower vs. upper cases, numbers: 1 and one'
Please use the chat interface to make it more interesting	The end navigation on (state) optional-profile subjects has not led to the database yet	Some abbreviations commonly used by students should be understood such as 'matkul' for 'mata kuliah' (course)
Use the university logo to fill the blank space	The system should direct users to answer a specific question and lead to the next steps	Requirements for taking KP and KKN should be added as a shortcut
There are still bugs in dialogue for taking subjects on semester I	It should be possible to go back to rewrite inputs due to misspelling or mistakes	The possibility of using abbreviations for optional profile subjects: SuLe for Supervised Learning

lot of qualified feedbacks to have a successful system improvement and IT students can provide such feedbacks. Due to limited space, the researchers exemplify five pairs of feedbacks gathered during these phases. Table II presents tester's feedback on interface design, dialogue navigation, and dialogue content.

The goal of system assessment is to evaluate the chatbot performance in having a chat with users. About 15 IT students are involved in this evaluation. 13.2% of them have been involved in the previous evaluation. In detail, 6.6% of the students have been involved in the first two data collection processes, while 6.6

Akin to the process of data collection, the researchers provide a task scenario to testers before they perform the chatting. Testers are asked to read the task scenario which consists of four tasks. Those are as follows.

- Task 1: having a dialogue on the number of credits which traverses the states 0,1,2,3, back to 0 or jump to 17 (see Fig. 3 for the state traversal)
- Task 2: having a dialogue on the subject offered which needs to traverse the states 0,4,5,6, back to 0 or jump to 17 for querying the compulsory subjects. As an option, testers can have a chat on the optional subjects, which need to traverse the states 0, 4, 7,8|9|10, 11|12|13|14|15|16 then to 17 or back to 10 and 7 or 0, or jump to 17.
- Task 3: querying the requirements of KP as a shortcut traversing the states 0, 18, 0|17
- Task 4: querying the requirements of KKN as a

TABLE III  
A SUMMARY OF STATISTICAL DATA ON THE SYSTEM ASSESSMENT PROCESS.

Testers Initials	Number of sessions	Number of dialogues	Numbers of utterance pairs per session
Zf	1	8	45
Dn	1	5	23
Eg	1	5	28
Ys	1	6	27
Al	1	4	43
Am	1	6	25
Ar	1	5	27
De	1	8	35
Hk	1	3	18
Ha	1	6	29
Ma	1	5	26
St	1	5	39
Af	1	2	15
An	1	6	39
Pa	1	6	30
Total numbers	15	80	449

shortcut traversing the states 0, 19, 0|17.

In having a chat with the chatbot, the testers perform various chat flows and dialogues. For a dialogue and session definition, the researchers follow Refs. [14, 20]. A session of chat refers to a tester's interaction with the chatbot within a given time frame. One chat session may comprise several dialogues. Those are a collection of user-chatbot conversation in which a user has succeeded in achieving the goal of conversation to get the information needed. Figure 5 illustrates one complete dialogue. Meanwhile, Fig. 6 exhibits the system has not delivered a partial dialogue since the number of credits being asked. The minimal number of dialogue done by testers achieves two, and the maximal number of dialogue achieved by some testers reaches eight dialogues in one session.

A written utterance refers to one complete sentence or linguistic fragments in a conversation which is typed by a user or prompted by the bot. An adjacency pair of utterances or a discourse in Foucault's terminology [14] marks a tester transition to different states. Thus, the number of utterances in one dialogue shows the flow of dialogue. In other words, it shows how testers traverse back-and-forth the states. Table III summarizes the number of sessions, dialogues, and written utterances in the process of system assessment.

In this evaluation process, the researchers record all testers' movement and behavior on the screen using Camtasia software as in the data collection process. However, following the chat, no questionnaire is administered to testers.

### B. Evaluation Metrics

One of the issues of developing a chatbot is in selecting the evaluation metrics to quantify system performance. Reference [2] listed several metrics from different perspectives in evaluating a DS. Information retrieval perspective will evaluate the system effectiveness by measuring precision, recall, and F-score. In user experience perspective, the goal of the bot is to maximize user satisfaction. Hence, bots are evaluated through questionnaires which rank it based on usability and user satisfaction [2]. In the linguistic perspective, bots should be evaluated on their ability to generate full, grammatical, and meaningful sentences. The used metrics are Word Error Rate (WER), Sentence Error Rate (SER), Concept Error Rate (CER), and Understand Error Rate (UER) [20].

Most mentioned metrics focus on evaluating the Spoken Dialogue Systems (SDSs) since they concentrate on the speech recognition and understanding. In the case of this bot which is based on written dialogue, such metrics do not apply well. Furthermore, some metrics offered by linguistic and information retrieval perspectives do not evaluate the effectiveness of an overall dialogue. Instead, they are applied on a per-utterance basis. For this reason, the researchers turn to dialogue-based metrics introduced by Ref. [20] and which have also been applied in [14].

The dialogue-based metrics measure the collective performance of the recognition, understanding, discourse and dialogue components [20] through Query Density (QD) and Concept Efficiency (CE). A concept, in this context, refers to a semantic unit realized as a keyword. For instance, in user utterance "How many credits can I take for the next semester if my semester GPA is 3.2?" There are two concepts in this utterance, namely credits and 3.2 of semester GPA. Although one keyword is realized in different word form, they will be counted as one concept if they refer to the same semantic unit.

QD measures how effective users can provide new concepts to the system by computing the mean number of a new concept introduced per-user query. It is computed by

$$QD = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{N_u(i)}{N_q(i)}, \quad (1)$$

where  $N_d$  is the number of dialogues,  $N_q(i)$  is the total number of user queries in  $i$ th dialogue, and  $N_u(i)$  is the number of unique concepts understood by the system in  $i$ th dialogue. A concept in a dialogue is not counted in  $N_u$  if the system had already understood it from a previously written utterance in one dialogue.

CE computes the average number of turns (similar to

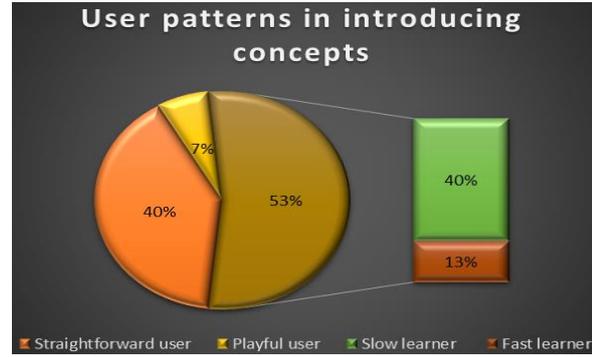


Fig. 7. Categorization of user behavior in introducing concepts during their chats with the chatbot.

a pair of utterances written in a reciprocal) necessary for each concept to be understood by the system [20]. CE is computed by

$$CE = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{N_u(i)}{N_c(i)} \quad (2)$$

where  $N_c(i)$  is the total number of concepts in  $i$ th dialogue. A concept is counted whenever it is written by users and is not understood by the system. Since,  $N_c(i) \geq N_u(i)$ , then  $0 \leq CE \leq 1$ .

### C. Results

The researchers conduct two kinds of evaluations concerning the experiment, the qualitative and quantitative evaluations. The researchers base the qualitative evaluation on the recorded dialogues between users and the chatbot. The researchers observe users' tendency in introducing new concepts which are shown in Fig. 7. Since the researchers do not evaluate user-experience perspective, the researchers base the analysis solely on the recorded discourses.

Based mainly on user utterances in discourse opening, it can be concluded that most users have a strong mental model on the QA chatbot. This is because they have been familiar with Siri or Google Now. As a result, 53% of users tend to introduce several concepts or keywords in one utterance. However, 13% of testers or users in this group can be identified as fast learners, as they introduce several concepts on the discourse opening of their first dialogue only. In contrast, 40% of them are classified as slow learners since they repeat this tendency in more than a half of their succeeding dialogues. This can also be interpreted that they apply their mental model to QA system for interacting with an FST-based chatbot.

40% of testers can be classified as straightforward users as they write their responses in phrases or sentence fragments. They obediently response as guided

TABLE IV  
DATA FOR QUERY DENSITY AND CONCEPT EFFICIENCY COMPUTATION.

Testers Initials	Number of dialogues	$\sum(N_u/N_q)$	$\sum(N_u/N_c)$
Zf	8	6.88	8.00
Dn	5	4.60	4.71
Eg	5	4.69	4.75
Ys	6	4.73	6.00
Al	4	2.40	3.36
Am	6	5.10	6.00
Ar	5	4.33	4.58
De	8	5.28	7.69
Hk	3	2.63	3.00
Ha	6	5.05	5.83
Ma	5	4.25	4.00
St	5	3.69	4.49
Af	2	2.66	1.82
An	6	3.58	4.87
Pa	6	4.44	5.62
Total numbers	80	64.37	75.73
QD/CE		0.81	0.95

by the bot prompts. As its consequence, this type of users has no difficulties with the dialogue flow of an FST-based chatbot. Only some users (7%) fell in the category of a playful one. The researchers based this categorization on their utterances that personify the bot, and express fun in having a chat. Their utterances also reflect that despite their awareness on the limitation of system capability, they like to know how far the system can response their queries. The following is an example of utterance in which the user addresses the chatbot as ‘Min’, a common name for a Javanese man but it is also a nickname of the administrator. For example, the utterance is “*Saya bingung mau ambil mata kuliah apa, bisa bantu Min? (I am confused to take what course, can you help Min?)*”.

The quantitative evaluation of system performance is measured through dialogue-based metrics: QD and CE. Using Eqs. (1) and (2), the results of QD and CE computation are presented in Table IV. The QD is 0.805, while the CE of the system reaches 0.95. The high CE rate indicates the system recognition on user inputs. In other words, the higher the efficiency is, the fewer times a user has to repeat a concept. The rationale is the use of keyword spotting technique which still recognizes the needed concept on a given time, although a user introduces more additional and unnecessary concepts on that given state or time. The QD rate of this chatbot is more than satisfactory. It shows that a user can communicate the concepts to the system. To increase the QD rate, the data collection with system-in-the-loop paradigm should be conducted with more users and done in several iterations.

In regards to these evaluation processes, the researchers identify this chatbot strengths and weak-

nesses. The system suffers from the common drawback of FST-based chatbot in which a user should prompt for specific concepts one-by-one to achieve the goal. This system is unfit for users who are familiar with QA system, but it is very suitable for a straightforward type of users. Another drawback of the system is that it is prone to typo errors and unable to recognize the misspelled concepts. To improve it, a spelling correction module should be added to its NLU component.

Despite these drawbacks, this chatbot is smart enough in recognizing different illocutionary acts such as asking, giving the orders, and teasing as presented in Table I. The rationale is the use of the keyword-spotting technique which recognizes the concepts only and disregards the rest. The other strength is that the system still recognizes a typo in part of multi-words concepts. In cases which a single concept is defined using several words, a typo in one of these words will not affect system recognition. For example, in the concept defined as ‘*mata kuliah pilihan non prodi*’ (course selection of non-department), the word ‘*prodi*’ is misspelled to ‘*produ*’ as in ‘*pilihan non produ*’ (found in 4<sup>th</sup> dialogue of 2<sup>nd</sup> user in Table IV).

## V. CONCLUSION

Developing a chatbot becomes a challenging task when it is built from scratch and independent of any SaaS. However, this research has proved that the combined methods of keyword spotting technique for the NLU component, FST for the Dialogue Management, rule-based keyword matching for NLG, and the system-in-the-loop paradigm for system validation can produce an efficient chatbot. Assessed with Dialogue-based metrics, the chatbot capability in understanding concepts introduced by testers is quite high with the QD score of 0.80, and CE score of 0.946. The high rate of CE proves that users do not need to repeat their utterances several times to be understood by the systems.

In addition to quantitative assessment, the researchers also observe user behavior or patterns in introducing the new concepts to be understood by the chatbot. This evaluation proves that only 40% of testers need more extended time in adapting the chatbot dialogue flow and understand that a single concept should be given in response to the system prompts. The researchers identify this group as ‘slow learner’ since their mental model on a QA system strongly influence their conversation with the chatbot. However, 60% of testers have no difficulties in interacting with the chatbot since they fall both in the category of fast learner or straightforward user.

The chatbot’s capability and flexibility in managing dialogues with users can be enhanced by using it as

a prototype to collect more data in the system-in-the-loop paradigm. The result of data collection can be used to build a conversational corpus which will enable the use of statistical or machine learning methods for improving the chatbot intelligence. Another alternative for future work is to apply the frame-based method in a component of DM. Then, to improve the chatbot capability, the future researchers can broaden the topic of conversation such as the subject description or summary.

#### REFERENCES

- [1] D. Allison, "Chatbots in the library: Is it time?" *Library Hi Tech*, vol. 30, no. 1, pp. 95–107, 2012.
- [2] J. Cahn, "CHATBOT: Architecture, design, & development," 2017, University of Pennsylvania.
- [3] J. Constine and S. Perez. (2016) Facebook Messenger now allows payments in its 30,000 chatbots. [Online]. Available: <https://techcrunch.com/2016/09/12/messenger-bot-payments/>
- [4] kata.ai. (2018) Studi kasus chatbot: Pelajari bagaimana chatbot membantu perusahaan mencapai tujuan bisnisnya. [Online]. Available: <https://kata.ai/case-studies/>
- [5] A. Bartl and G. Spanakis, "A retrieval-based dialogue system utilizing utterance and context embeddings," in *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, Dec.18–21, 2017, pp. 1120–1125.
- [6] N. M. Radziwill and M. C. Benton, "Evaluating quality of chatbots and intelligent conversational agents," *ArXiv Preprint ArXiv:1704.04579*, 2017.
- [7] M. L. McNeal and D. Newyear, "Introducing chatbots in libraries," *Library Technology Reports*, vol. 49, no. 8, pp. 5–10, 2013.
- [8] M. McTear, Z. Callejas, and D. Griol, *The conversational interface: Talking to smart devices*. Switzerland: Springer, 2016.
- [9] T. Klüwer, "From chatbots to dialog systems," in *Conversational agents and natural language interaction: Techniques and effective practices*. IGI Global, 2011, pp. 1–22.
- [10] 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, pp. 72–80, 2015.
- [11] A. Niklasson, "Dialogue systems using web-based language tools," Master's thesis, Umea University, Sweden, 2017.
- [12] S. Ghose and J. J. Barua, "Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor," in *International Conference on Informatics, Electronics & Vision (ICIEV)*. Dhaka, Bangladesh: IEEE, May 17–18, 2013, pp. 1–5.
- [13] B. Setiaji and F. W. Wibowo, "Chatbot using a knowledge in database: Human-to-machine conversation modeling," in *7th International Conference on Intelligent Systems, Modelling and Simulation*, Bangkok, Thailand, March 16, 2016, pp. 72–77.
- [14] L. D. Krisnawati, "Implementing mixed initiative dialogue in voiceXML to process CIS queries," in *Seminar Nasional Sistem dan Informatika*, Bali, Nov. 16, 2007, pp. 149–155.
- [15] M. Dahiya, "A tool of conversation: Chatbot," *International Journal of Computer Sciences and Engineering*, vol. 5, no. 5, pp. 158–161, 2017.
- [16] F. Morbini, E. Forbell, D. DeVault, K. Sagae, D. R. Traum, and A. A. Rizzo, "A mixed-initiative conversational dialogue system for healthcare," in *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Seoul, South Korea: Association for Computational Linguistics, July 5–6, 2012, pp. 137–139.
- [17] D. Griol and J. M. Molina, "A proposal to manage multi-task dialogs in conversational interfaces," *Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ)*, vol. 5, no. 2, pp. 53–65, 2016.
- [18] M. Hirzel, L. Mandel, A. Shinnar, J. Simon, and M. Vaziri, "I can parse you: Grammars for dialog," in *2nd Summit on Advances in Programming Languages (SNAPL)*, Yorktown Heights, USA, May 9, 2017, pp. 6:1–6:15.
- [19] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, and Z. Li, "Building task-oriented dialogue systems for online shopping," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017, pp. 4618–4626.
- [20] J. Glass, J. Polifroni, S. Seneff, and V. Zue, "Data collection and performance evaluation of spoken dialogue systems: The MIT experience," in *Sixth International Conference on Spoken Language Processing (ICSLP)*, 2000, pp. 1–4.
- [21] M. Gabsdil, "Clarification in spoken dialogue systems," in *Proceedings of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*, 2003, pp. 28–35.