IPTEK The Journal of Technology and Science, 34(3), 2023 (e/pISSN:2088-2033, 0853-4098)

DOI: 10.12962/j20882033.v34i3.16869 | Received 27 May, 2023; Revised 27 May, 2023; Accepted 6 July, 2023

ORIGINAL RESEARCH

HYBRID HUMAN-CONTEXTUAL CHATBOT WITH CONTINUOUS MODEL DEVELOPMENT USING RECURRENT NEURAL NETWORK FOR HELP-DESK SUPPORTING TOOL

Dyah Ayu Permata Sari | Ahmad Saikhu* | Ratih Nur Esti Anggraini

Departement of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Correspondence

Email: saikhu@if.its.ac.id

Present Address

Magister Tower, Jl Informatika No. 10, Surabaya 60111, Indonesia

Abstract

The development of chatbots is currently quite significant, considering the trend of interactive customer services 24/7. One advantage of using chatbots is reducing queues and customer waiting times. However, previous research stated that 87% of users prefer interaction with humans to solve complex problems. Therefore, this study introduced a hybrid human-contextual chatbot with sustainable model development using Recurrent Neural Network (RNN) and threshold optimization. The research proposes a cooperation framework between Artificial Intelligence (AI) and humans to optimize the workforce while maintaining the quality of the company's services. The system has a monitoring website and continuous model development to ensure the continued growth of the model. The system trial was conducted in XYZ company's IT management division in Indonesia for four weeks. The performance evaluation process uses accuracy, hand-off rate, average execution time, and average response time. Weekly performance evaluation results obtained accuracy and hand-off rate score increased, but average execution time and response time decreased every week. The decrease in execution and response time indicates a faster model performance. The highest accuracy and hand-off rate values were 0.99 and 0.98, respectively. Execution and response times get the lowest seconds at 0.39 seconds and 0.85 seconds, respectively.

KEYWORDS:

cooperation framework, supporting help-desk tool, threshold optimization.

1 | INTRODUCTION

Chatbots are generally installed in messaging applications for information dissemination and specific communication tasks. The development of chatbots is currently quite significant, considering the trend of interactive customer services 24/7[1]. Chatbots are widely used in sales (41%), customer service (37%), and marketing (17%)[2]. The great interest in using chatbots is based on their superiority in reducing queues and customer waiting times. This aligns with the company's efforts to achieve efficiency and reduce labor costs[3]. Service efficiency and effectiveness must also be ensured not to reduce the value of customer satisfaction. Successfully resolving complaints is also a challenge that cannot be ruled out. Many ideas arose to build a tool to understand customers' thoughts, feelings, and expected results[3]. Researchers have proposed the development of artificial intelligence in chatbot development to optimize capacity and service quality. Because understanding the language contextually is also needed in communicating besides using the language itself[4]. Comprehending the meaning of sentences encourages contextual-based chatbot development. Contextual understanding is fundamental for better problem-solving abilities[4].

Sentences are composed of several words, and each word's meaning can influence one another. So solving problems in text data generally requires understanding the meaning of all the words in the sentence[5]. Understanding the meaning of each word in a sentence is called contextual understanding[6]. Therefore, this study used the Recurrent Neural Network (RNN) method to build a contextual model. RNN can store previous data memory for processing each sample by considering prior data. This makes the RNN method widely applied to solve the problem of sentiment analysis and classification of text data. In this study, the RNN method was chosen as a model for classifying data sent by users via chatbots. The selection of RNNs was also based on the case studies taken, that is helping the help-desk to answer questions frequently asked by users. These questions tend to be in sentences that are not too long, and the communication between the user and the help-desk is in short conversations. The RNN model's simple architecture also benefits the speed of generating predictions, thereby reducing customer waiting times.

The development of chatbots equipped with artificial intelligence has caught the attention of many companies[7]. However, consumer preferences for resolving complaints with the help of chatbots are still not in line. Research by Kirkpatrick stated that 87% of consumers prefer interaction with humans. This statement is based on a customer's assessment of the chatbot's ability to be lacking in dealing with complex problems[8]. From this perception, it can be said that processing customer complaints cannot be left entirely to the chatbot. Human intervention is still needed to solve complex problems[9]. A framework that accommodates cooperation between humans and bots to fill each other's strengths and weaknesses is proposed in this study. The hybrid human-contextual chatbot framework is a flow of collaborative activities between humans and bots to support continuous model development. This framework has a threshold value for each class so that the help-desk team can participate in dealing with more complex problems. Two scenarios may occur after the results of the contextual model predictions are successfully obtained. In the successful scenario, the system sends a response that matches the predicted class that has been generated. In an unsuccessful scenario, the system contacts the help-desk team to solve the reported problem.

A good framework is expected to provide a clear picture of each party's responsibilities to create labor efficiency. Harjanto and Mahendrawathi's research states that to improve organizational performance, an organization needs to improve the practice of strategic view, performance measurement, and process definition[9]. Collaboration systems between humans and bot agents can also offer convenience in maintaining the quality of company services. The quality of problem-solving by agents with artificial intelligence is also a topic of much discussion. Various approaches to measuring response quality were introduced from quantitative evaluations by distributing questionnaires to evaluation matrices, such as accuracy, precision, and F1 value. This study proposed two performance evaluation references from the application and model sides. The average execution time and accuracy values were selected as model performance evaluation parameters. Otherwise, the average response time and hand-off rate are chosen as application evaluation parameters. The execution and response time have also been widely used as evaluation methods in automation systems with artificial intelligence[10],[11]. The evaluation process is carried out to provide an overview of the capabilities of the proposed approach in helping the sustainability of business processes in a company. This paper is structured as follows. Section 2 provides information related to previous research and the research gap. Section 3 describes the methods used to design and build the proposed system. Section 4 presents the results of the method implementation. Section 5 describes the results of testing the system in real company case studies in Indonesia. Section 6 lays out the conclusions from the trial and suggestions for future research.

2 | PREVIOUS RESEARCHES

RNNs are widely used in text data and language processing. The different characteristics of each language are a challenge in itself.[12] conducted sentiment analysis with BiLSTM on datasets in Bengali, and[13] performed a sentiment analysis with LSTM, GRU, Bi-LSTM, and Bi-GRU models on Amazon's review dataset. While in this study, using frequent ask questions in Indonesian. Preliminary data processing is required under Indonesian language processing rules. Some studies also add personality features to the model.[14] classify hateful content with the user's information feature in the Ensemble Recurrent Neural Network.[15] conducted a classification of malware using RNNs and three different feature vectors. The structure and relationship of each training data have a significant impact on the reliability of the model. The broader application of chatbots encourages researchers to analyze the impact and effectiveness of chatbots.[16] conducted a quantitative test regarding customer trust in automated chatbot services.[17] investigate chatbots' problem-solving skills and task complexity. Objectively, this research aims to examine consumer trends toward AI and humans. It is concluded that in tasks with low sophistication, consumers consider AI. Conversely, in the case of tasks with high levels of complexity, consumers prefer humans as a solver. In addition, problem-solving abilities also affect customer intentions in using the service. Based on previous research, human intervention is still needed in the case of complaint resolution services. The need for work automation and workforce optimization is what the company strives to achieve. Therefore, this study introduces a framework that combines automation with certain limitations for human assistance. The companion here aims to help humans deal with repetitive and manageable problems that do not require intervention from other parties. The limitation of contextual chatbot services must be met with the predefined condition that compares the threshold and the model's predicted results. The prediction probability is the result of model predictions other than tags or data classes. The prediction probability for each class is generated by calling the softmax activation function. The importance and impact of the threshold value make the threshold optimization carried out by many previous studies.[18] perform multi-class data classification with threshold value optimization using ROC and the confusion matrix approach. The proposed threshold optimization in this study was carried out with TPR and FPR values using the OvR approach. TPR and FPR are obtained from the ROC function with the help of the scikit-learn library. However, the approach to obtaining the optimal threshold value is slightly different. Where is[18] using a confusion matrix, while this study uses the One vs Rest (OvR) approach to get the optimal threshold. Using predefined conditions with threshold values and predicted probability creates a cooperative framework between bots and humans. The analysis process after chatbot development has received much attention. Development frameworks and quality assurance after the launch of chatbots have been proposed by many researchers before.[19] proposed a methodology for content management in chatbot systems called Chatbot Management Process (CMP). Santos et al. aims to develop chatbot content through the analysis of user interaction, thus enabling the creation of a human-supervised work cycle. CMP is intended to maintain and safeguard the content of the chatbot with machine learning after its launch.[20] proposed a Knowledge-Based system based on knowledge graphs via Linked-data called KBot. Evaluation results with precision, recall, and F-measure values show that KBot performs well. The limitations of previous research have not accommodated collaboration between a human professional and a bot agent as artificial intelligence. Many previous studies have focused on improving system performance with various model modifications. On the other hand, a business framework for continuously transferring human knowledge to artificial intelligence agents has not been introduced. The process of adding, maintaining, and providing indicators in determining training data is also an important issue to discuss. Therefore, this research proposes a framework for cooperation between humans and artificial intelligence agents using threshold optimization and the RNN method. The proposed framework also supports continuous model development. The model development process is based on observing the previous model's capabilities. The monitoring website was built to provide actual data from system performance. Likewise, performance evaluation and model building are carried out periodically. The evaluation matrices used in this study include accuracy, hand-off rate, average execution time, and average response time.

3 | METHOD

This chapter describes the stages in the research conducted. The process starts with data collection, design and development, testing, and evaluation.

FIGURE 1 Relationship between patterns, tags, and responses growth chart.

TABLE 1 Training data details

Attribute	Description	Data Type	Example
Pattem	Frequently asked questions received by the help-desk	Text	Why can't I log in and just information appears continuously processed.
Response	The appropriate response for each pattern created	Text	Please check the URL you are going to Delete all text after the main address.
Tag	Decision classes provided by the help desk team	category	err login

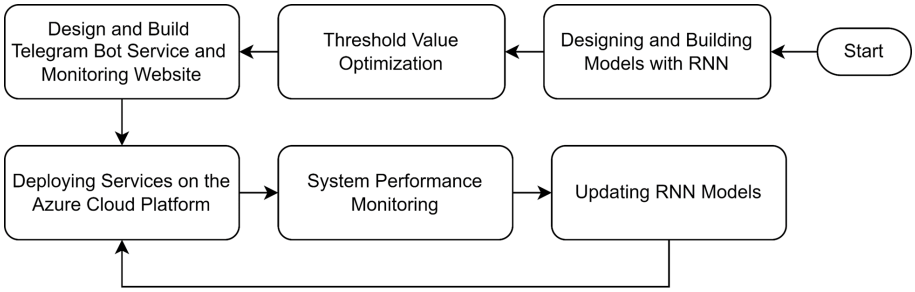

FIGURE 2 Design and development flow for Hybrid Human - Contextual Chatbot.

3.1 | Data Collection

The data collection process is carried out to obtain the primary model. The data used in this study is a list of frequently asked questions received by the help-desk team three months before. Designing and building a basic model requires training data with three main attributes: pattern, tag, and response.

Frequently asked questions from the help-desk team will be treated as pattern data. Furthermore, the help-desk team is asked to provide a tag and appropriate response for each pattern. The relationship between patterns, tags, and responses is described in Figure 1 . Tags are like class data for patterns and responses. Each response and pattern must have one tag, and each tag must have at least one pattern and one response. The characteristics of the data can be seen in Table 1 .

3.2 | Design and Development.

The system design and development process starts from model building to deploying chatbot services and website monitoring. This process still requires further steps in monitoring system performance and updating models regularly to support sustainable model development. A series of design and development processes are shown in Figure 1 . The system built uses primary data and is applied to real case studies of Informatic Technology company in Indonesia. Two main systems were developed in this research, namely the chatbot service and the performance monitoring website. These two services are interconnected to support a continuous model development process. The chatbot service in which an artificial intelligence model is built using RNN, and its performance is monitored through the website. Monitoring performance and updating training data are done with human assistance. This is intended to guarantee the response's quality. The performance monitoring website provides data on the user's question, chatbot's answer, predicted probability, predicted tag, last threshold value, prediction time, and response time. Through this data, the help-desk can carry out the process of monitoring performance and updating training data. This

process results in the growth of the training data and the improvement of the model's ability to predict future data. The mapping of the design and development stages can be broadly divided into three parts: RNN model development, threshold optimization, and system deployment. Developing the RNN model begins with pre-processing data. The NLTK (Natural Language Toolkit) library assisted with tokenization, and the Sastrawi library helped stemming. Stemming is the process of getting the primary word. The next stage is the removal of Indonesian stop words and non-standard terms. From the series of processes carried out, it will produce primary word lines that play a role in building contextual models. Before creating a contextual model with the RNN method, it is necessary to vectorize the data. The Bag of Words (BoW) method converts unique primary word rows into vectors. This vector will be the input for developing the basic model. The PyTorch library was used to build the RNN model. After the construction of the basic model has been successfully carried out, the next step is to find the threshold value. Multi-class classification uses model-generated probability comparisons to determine predictive class outcomes. The way it works is to compare the probability values generated by the model for each class. The class with the highest probability will be the predicted result. Therefore, the search for threshold values in this study uses the One vs Rest (OvR) approach. OvR is a model evaluation technique that compares each class to other classes. It is implemented by taking one class labeled as positive while the rest are considered negative. OvR was chosen because of the efficiency and character of the research data, which has many classes with the same importance value. The ROC curve is used to determine the optimal threshold value. ROC Curve is a curve that illustrates the diagnostic capabilities of systems in binary classification. This curve shows the classifier behavior for each threshold with two variables: True Positive Rate (TPR) and False Positive Rate (FPR). The TPR and FPR of each tag are used to produce the optimal threshold value. Optimizing threshold value is to improve accuracy and TPR [18]. It is necessary to record the threshold value for each model development. The threshold value is needed as a comparison value under predetermined conditions.

3.3 | Design and Development.

Chatbot is built on the Telegram social media platform to reach issues reported by users. Integration is carried out between the telegram bot service and the RNN model for calling and selecting answers based on models' predictions. Another service that needs to be built is website performance monitoring. Performance monitoring website is used to evaluate model and system performance. All activities involving chatbots are recorded in the MySQL database and displayed on the monitoring website. Django was chosen as the website development framework for the performance monitoring website. Performance monitoring website also functions as a system administration service, so several other modules are built into it. The supporting modules are the master data model, help-desk contacts, dialogue log, threshold log, and connection log. Then the two services are deployed on the Azure Virtual Machine to be used by users in real case studies.

3.4 | Testing

A hybrid human-contextual chatbot framework is shown in Figure 3 . System testing begins with users asking questions to chatbots on Telegram. Questions given by the user become input to the RNN. The model generates probability and predictive tags. The next step is to compare the prediction probabilities with the threshold values of the same tags. When the model produces a predicted value less than the threshold, the chatbot sends a connection request to the help-desk that is not connected to any connection. The help-desk needs to approve requests to communicate with users so that the chatbot can establish a connection between them. After establishing a connection, the chatbot sends the user's last conversation data to the connected help-desk. The help-desk team continues to solve problems that chatbots have not successfully resolved. The framework provides behavioral standards that can be implemented to end the relationship between the help-desk and users via chatbots. Before the session ends, the help-desk confirms whether the solution has answered the reported problem. The session can be terminated if the user states that the solution has solved the reported issue and help-desk sends '/finish' command.

3.5 | Evaluation

The evaluation process is based on data obtained from the monitoring website. Evaluations occur every week or seven days, equal to the model development process. Accuracy and execution time become the model performance evaluation. Otherwise, the hand-off rate and the response time become the application performance evaluation. The execution time is obtained from the model's time to generate the appropriate tag, while the response time is the required time for the system to respond to the user. Starting from a model that makes predictions and then a chatbot that looks for an appropriate response and sends it to the user. Accuracy is obtained from the number of questions answered correctly divided by the questions successfully answered

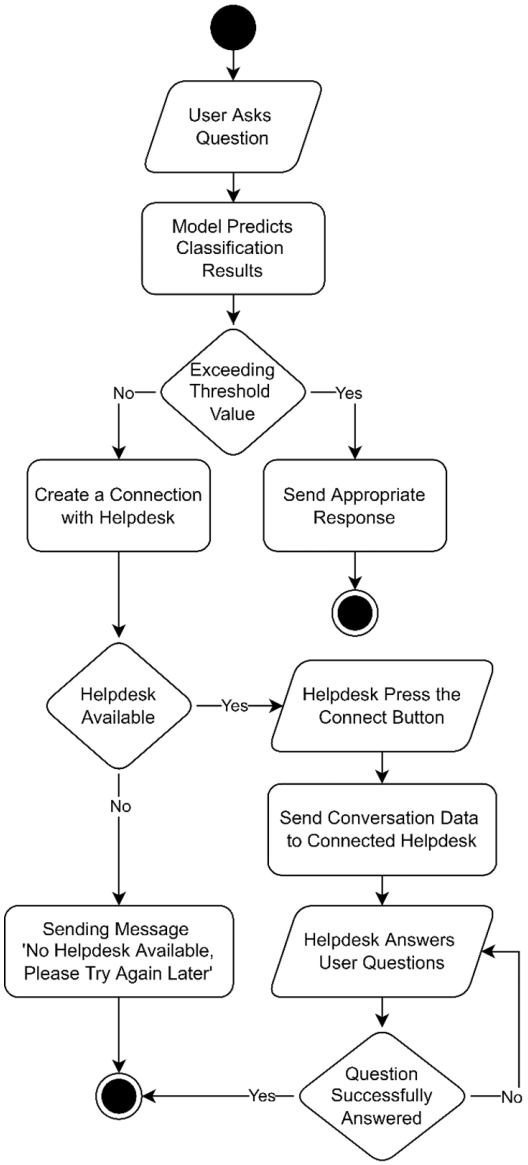

FIGURE 3 Hybrid Human-Contextual Chatbot business framework.

by the chatbot. On the other hand, the hand-off rate is obtained from the number of questions answered correctly divided by all questions submitted by the user in a unit of time. The accuracy and hand-off rate formulation are written in Equation 1 and Equation 2, where N denotes the amount. The development of the system performance can be seen from the increased accuracy and hand-off rate value every week. In contrast, the application performance can be seen from the decrease in the average execution and response times. The hand-off rate value indicates the percentage of help-desk tasks successfully replaced by the proposed system. A high hand-off rate indicates the system's success in helping the help-desk complete its tasks. Help-desk can hand over the task of solving low-complexity problems to bots. On the other hand, highly complex complaints can be handled by the help-desk team. The evaluation process is needed to answer the system's ability to maximize employee performance and company services.

$$Accuracy = \frac{N\,RespondedCorrectly}{N\,QuestionAnswered} \tag{1}$$

$$Hand - off\,Rate = \frac{N\,RespondedCorrectly}{N\,QuestionAnswered} \tag{2}$$

4 | IMPLEMENTATION

This chapter describes system development results from implementing the previous discussion. It includes the results of developing two primary services: chatbot and website services.

4.1 | Data Processing

This study used data on questions frequently obtained by the help-desk team in the last three months at XYZ company. The learning data for the first model totaled 408 lines of question data, 24 data classes, and 24 types of answers. Each class has one answer and many possible questions. The data obtained is mapped to meet the specified intent data structure, as shown in Figure 1. Data intent mapping is done by making data classes as tags, answers as responses, and every frequently asked question as a pattern. The mapped data is then entered into the MySQL database. The use of databases is needed for ease of storing, accessing, and modifying data. The system built supports the development of the model on an ongoing basis. Increasing the number of tags, responses, and patterns is possible when testing the system on real case studies.

4.2 | Model Training

RNN model training supported by the PyTorch library. The model has two hidden layers, 128 hidden sizes, and one fully connected layer. The fully connected layer converts the RNN output into the desired output form. The training data used 1000 epochs and a 0.001 learning rate. Epoch is a parameter used to determine how many times the RNN algorithm works through the entire dataset, both forward and backward. Meanwhile, the learning rate is a data learning parameter for calculating weight correction values. This study also uses CrossEntropyLoss for the loss function and Adam for the optimizer function. The use of CrossEntropyLoss as a loss function is due to its ability to summarise the average difference between the actual and predicted probability distributions for all classes. The last step for model training is saving the model and ensuring the model can be accessed and used in the chatbot service.

4.3 | Getting Optimise Threshold

A threshold is needed for a cooperation framework between bots and humans. The optimal threshold value can help increase the hand-off rate. ROC Curve with the OvR approach is used to obtain the optimal threshold value in this study. The OvR approach is widely chosen for the case of multi-class data classification. Make the OvR method suitable for training data with more than 24 tags. Using the OvR approach produces threshold values for each available data class. This happens because of the way OvR works, which performs a binary classification for each class with the rest of the classes.

$$TPR = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3}$$

$$FPR = \frac{FalsePositive}{TrueNegative + FalsePositive} \tag{4}$$

ROC curve shows the classifier behavior for each threshold with the True Positive Rate (TPR) and False Positive Rate (FPR). TPR is obtained from the number of positive data that are predicted correctly (True Positive) compared to the number of positive data that are predicted correctly (True Positive) plus the amount of negative data that is mispredicted (False Negative), as defined by Equation 3. In comparison, the FPR value is obtained from a comparison of the number of positive class data that is incorrectly predicted (False Positive) with the amount of negative data that is correctly predicted (True Negative) plus the amount of positive data that is incorrectly predicted (False Positive), as defined by Equation 4. A good Area Under Curve (AUC) value is obtained from a high TPR value and a low FPR. The value of TPR or sensitivity and TNR or specificity have the relationship shown in Equation 5.

$$TNR = 1 - FPR \tag{5}$$

$$J = TPR + TNR - 1 \tag{6}$$

$$T_{\text{opt}} = org_t max(TPR - FPR) \tag{7}$$

The search for optimal threshold values uses Youden's J statistics approach[21]. Youden's J statistic is a single statistic that captures the performance of a dichotomous diagnostic test. It is determined that the optimal probability value is obtained from sensitivity + specificity – 1, shown by Equation 6. The equation is adjusted to Equation 5 and becomes Equation 7, which can be used to find the optimal threshold value. Based on Equation 7, the optimal threshold value is the maximum index value obtained from (TPR - FPR).

4.4 | Website Development

Django was chosen as the framework for website data modeling and system performance monitoring. The website can be accessed via the XYZ company intranet. The access rights that can access and use the monitoring website are the admin and members of the help-desk team. The modules contained in the website monitoring service include dialogue logs, connection logs, threshold logs, and master data. The Dialog Log module is intended to record every conversation a user has with a chatbot. Model and system performance logs are also stored in this module. Dialogue Log data includes the date of the conversation, the questions given by the user, the answers provided by the chatbot, the prediction probability value, and so on. Through this data, the administrator can monitor system performance. User questions that the chatbot cannot understand are forwarded to the available help-desk. The connection between the help-desk and the user is recorded in the Connection Log module. The definition of an available help-desk is a help-desk that is not connected to any user. If no available help-desk is found, the system sends a message to try again later because the help-desk is unavailable. This module aims to help share the workload between help-desk members. The Threshold Log module is used to record changes to threshold values. The proposed framework uses threshold values to support the collaboration between humans and chatbots. The threshold value is the minimum predicted probability value that must be achieved. If the model cannot obtain a prediction percentage, the question will be forwarded to the help-desk team. The development of model training data allows changes in the optimum threshold value for each available class. Therefore, records of changes to threshold values need to be kept. There are four modules for master data, namely help-desk contact, tag, pattern, and response. The identity of the help-desk members is recorded in the help-desk contact module. This is necessary for connection requests. Furthermore, there is a Tag module for recording training data labels. In comparison, the Pattern module is a recording of questions that the user may ask. The list of possible and frequently asked questions is collected by the help-desk team and mapped with the appropriate tags. Next, the Response module is a recording of the proper answers.

4.5 | Chatbot Development

Chatbot is built on top of the Telegram social media platform to reach and make it easy for users. The model and the chatbot service are integrated for calling, predicting, and selecting answers. The development of the telegram bot service uses the Python programming language and is assisted by the python-bot-telegram library. Several commands can be executed in the chat room with the chatbot, namely /start, /finish, and approve connection requests by connect button. The initial command that can be sent to the bot is /start. This command is informative and is not required to be sent. The chatbot establishes a connection between the user and the help-desk team if the chatbot fails to understand the question. Connect requests are sent to the available help-desk. Help-desk needs to press the connect button to approve the connection request. After establishing a connection, the chatbot sends a conversation log to the connected help-desk. This function facilitates the help-desk to understand the problems reported without re-asking the user. The help-desk connected to the user can confirm the resolution of the reported problem. The help desk can send the command /finish to end the connection if the issues are resolved. In other words, the /finish command can only be done by the help-desk team. If the help-desk currently connected to the user does not send the /finish command, the system terminates the connection after 15 minutes without exchanging messages between the two.

5 | RESULT

This chapter describes the results of system implementation. The trial runs from 1st February 2023 to 3rd March 2023. System evaluation and model development are carried out in stages once a week. The stages of model development include recording system performance and adding training data. The help-desk team can monitor chatbot performance through the website. The help-desk team can design training data from this monitoring process to improve the model's predictive ability. A manual

TABLE 2 The growth of training data

Week	Total Data	Growth	Number of Classes
1^{st}	408	103	24
2^{nd}	511	122	28
3^{rd}	633	135	31
4^{th}	768	-	33

assessment is also carried out by the help-desk team regarding the response given by the chatbot. This ensures the model's predictive ability and threshold values work correctly. The accuracy value of the built model is obtained from the suitability of the response given by the chatbot, while the hand-off rate is obtained from the ability of chatbots to replace the role of a help-desk. The differences and calculations of the two can be seen in Equation 1 and Equation 2

TABLE 3 The system performance evaluation.

Matrices	1^{st} Week	2^{nd} Week	3^{rd} Week	4^{th} Week
Accuracy	0.93	0.96	0.97	0.99
Hand-Off Rate	0.79	0.88	0.93	0.98
Average Execution Time (sec)	0.52	0.52	0.49	0.39
Average Response Time (sec)	1.03	1.03	0.95	0.85

The system trial was carried out for four weeks at the XYZ company. At the same time, the administrator carries out the model development every Sunday or Monday morning. Table 2 describes the growth of training data during the system trial period. The data used to build the basic model consists of 408 training data with 24 tags. This number continues to grow until it is in the 4^{th} week. Training reached 768 data and 33 tags in the last week of trials. The results of the system performance evaluation with the four assessment criteria are shown in Table 3 . A comparison of model performance using other methods was also carried out in this study. Neural Network (NN) and Support Vector Machine (SVM) were chosen to build a contextual model from the training data obtained in system testing at XYZ company. A Neural Network with two hidden layers and SVM with Radial Basis Function (RBF) kernel is the architecture used. The results of testing the contextual model using the NN, RNN, and SVM methods are presented in Table 4 .

TABLE 4 Performance of the Contextual Model Method

Method	Accuracy (%)	Hand-off Rate (%)	Average Execution (secs)
Neural Network	0.94	0.91	0.37
Recurrent Neural Network	0.99	0.98	0.39
Support Vector Machine	0.97	0.25	0.33

6 | DISCUSSION

The evaluation is done weekly to see the model and system performance growth. Taking one week for performance observations is adjusted to the model development process. Four evaluation matrices are used to measure model and system performance: accuracy, hand-off rate, average execution, and average response time. Model accuracy is shown with a line graph that continues to increase weekly in Figure 4 . This can be due to the growth of training data and the continuous model development process.

The hand-off rate shows the chatbot's ability to help and replace the help-desk tasks. The main job of the help-desk is to provide the right action and response to the questions users ask. Based on Figure 5 , there is an increase in the hand-off rate value every week. This shows that the percentage of chatbot usability is getting better. The chatbot usability percentage is in line with the help-desk workforce optimization objective. The execution time is obtained from the model's time to generate the appropriate tag. The weekly average execution time caused by the model has decreased, although not significantly, in Figure 6 . This indicates that the model is also growing faster in generating predictions. The response time is the required time for the system to respond to the user. Starting from a model that makes predictions and then a chatbot that looks for an appropriate response and sends it to the user. The development of response time performance every week is shown in Figure 7 . The decreased average response time indicates a faster time to generate predictions. It can be said that adding training data without any changes to the model architecture can affect the time in generating predictions.

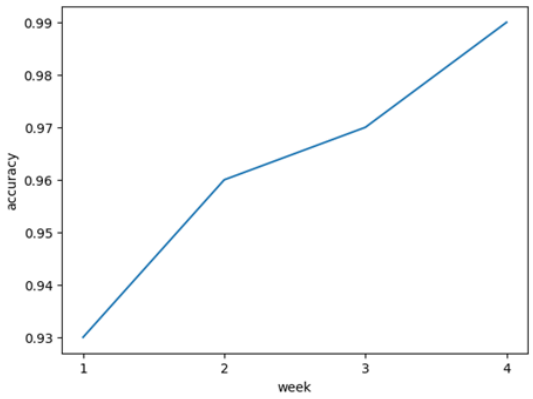

FIGURE 4 Weekly accuracy growth chart.

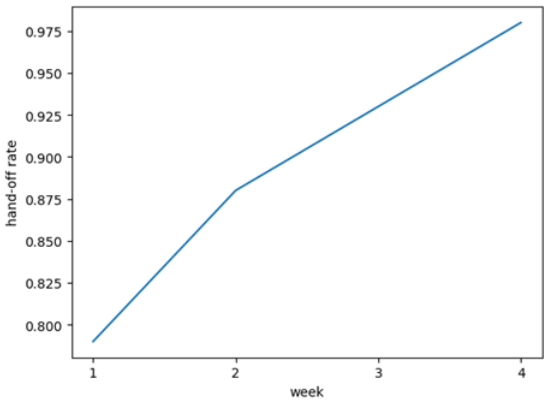

FIGURE 5 Weekly hand-off rate growth chart.

System performance growth in accuracy and hand-off rate occurs every week. In the first week, an accuracy value of 0.93 was obtained, and it continued to increase so that in the 4^{th} week, it became 0.99. In line with accuracy, the value of the hand-off rate also continues to grow. The hand-off rate was obtained in the first week at 0.79 and increased to 0.98 in the 4^{th} week. Unlike the accuracy and hand-off rate, the average execution and response time have decreased weekly. The average value of execution time in the first week is 0.52 seconds and continues to decline until 0.39 seconds in the 4^{th} week. On the average execution value, a similar pattern is also obtained. The average response time in the first week was 1.03 seconds, down to 0.85 seconds in the 4^{th} week. Models built using the NN and SVM methods were evaluated using the accuracy, hand-off rate, and average execution time. Model development uses training data obtained from system implementation at XYZ company. The data is processed to produce optimal threshold values according to the method proposed in the previous discussion. The testing process is carried out with data on questions submitted by users in the 4^{th} week of system implementation. The results of the contextual evaluation of the model using the NN, RNN, and SVM methods are shown in Table 4 .

The test results show the highest accuracy value is obtained by the RNN method, which equals 0.99. The accuracy produced by the NN method is 0.94, and SVM is 0.97. The model got the highest hand-off rate value with the RNN method, 0.98. In the model with the NN method, the hand-off rate is 0.91, and the SVM method is 0.25. The accuracy obtained by the SVM method is relatively high, but the hand-off rate is shallow. This is due to the low probability of the SVM method's predicted results, so chatbots create more requests connected to the helpdesk than answering user questions directly. In contrast to the value of accuracy and hand-off rate, the average execution time of the SVM method is the best. This indicates that the average time required for the contextual model using the SVM method to produce the fastest probability of the predicted result is 0.33 seconds. The average execution time for the NN method is 0.37 seconds, while for the RNN method is 0.39 seconds. This makes the RNN method with an architecture of two hidden layers, 128 hidden states, and one fully connected layer the longest to generate the predicted probability results.

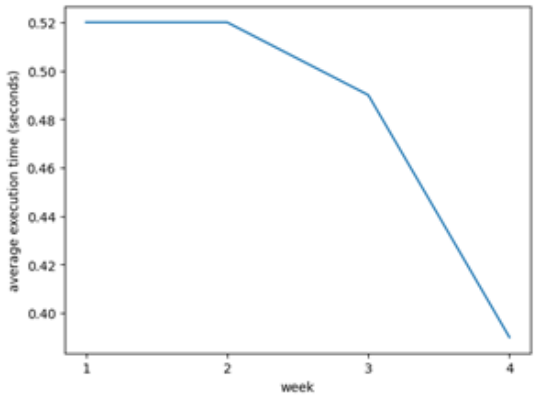

FIGURE 6 Weekly average execution time growth chart.

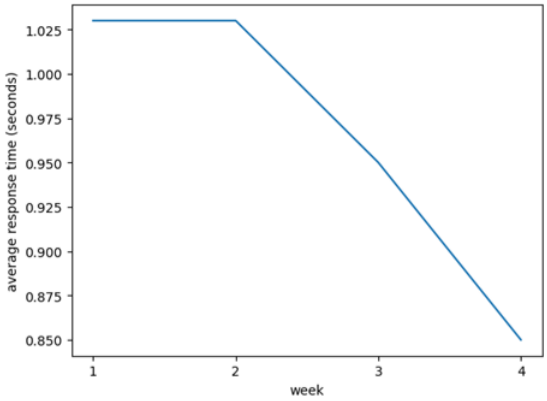

FIGURE 7 Weekly average response time growth chart.

7 | CONCLUSION

This study proposes a business framework for collaboration between humans and artificial intelligence (AI) agents. The hybrid human-contextual chatbot with continuous model development is presented to reduce waiting time and workforce optimization. The contextual chatbot is built on the RNN model to predict class answers. Threshold optimization with TPR and FPR values using the OvR approach is carried out to create intervention human flow. The monitoring website is built to monitor the performance of the model and application. The proposed system was tested for four weeks at the IT Division of XYZ company in Indonesia. Four evaluation matrices are used to measure model and system performance: model accuracy, hand-off rate, average execution time, and average response time. The framework supports the model development process on an ongoing basis so that the evaluation process is carried out every week. Taking a seven-day timeframe for performance observations is adjusted to the model development process in the field. In the last week of testing the system, an accuracy of 99% was obtained. The accuracy value indicates the model's ability to provide correct predictions for each user question. At the end of the study, the hand-off rate value obtained was 0.98. The hand-off rate value shows the ability of the system to replace the help-desk role in answering questions. So that the help-desk can focus on carrying out other tasks, such as solving complex problems, evaluating solver performance, and monitoring automation systems. This aligns with the research objectives: optimizing the workforce and maintaining the company's service quality. In contrast with the accretion of accuracy and hand-off rate value, the average execution and response time decreased every week. The decrease in execution and response time indicates faster the model's performance in generating predictions and the system in providing answers. The lowest average execution time and total response were found in week 4th, which were 0.39 seconds and 0.85 seconds, respectively. Performance testing using methods other than RNN is also carried out to prove the suitability of the methods used with the proposed framework. The performance comparison between the Neural Network, Recurrent Neural Network, and Support Vector Machine methods using a matrix of accuracy, hand-off rate, and total execution time. The test results stated that the contextual model obtained the highest accuracy and hand-off rate values using the RNN method. This shows that the RNN method in the cooperation framework between humans and bots as artificial intelligence is the right choice for XYZ company. Growing data and operational needs encourage the system to work according to company standards and cultures. Evaluation of system performance through the monitoring website has succeeded in assisting the help-desk in compiling quality training data. Growing training data gives the model and systems performance improvements in generating predictions and responses to users. A business framework supported by sustainable model development has proven effective and adaptive to future needs practically. Scientifically also confirms that the growth of training data on the same model architecture does not slow down the model's performance. The quality of the training data and its suitability for usage needs present a model with efficient weights and formulas. This makes the model faster in making predictions and increases the average execution and response time by decreasing the resulting time unit. In further research, the process of optimizing thresholds and model architecture can be carried out with more methods, such as Balance Accuracy for optimizing threshold values and Long Short-term Memory as a model contextual method. Learning and optimizing threshold values with various methods will increase system robustness. Adding a voice recording feature to communication between the helpdesk and users via the telephone can also be a breakthrough in better capturing model development needs.

ACKNOWLEDGMENT

CREDIT

Dyah Ayu Permata Sari: Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft, Visualization Ahmad Saikhu: Supervision Ratih Nur Esti Anggraini: Supervision

References

1. Vaios Koumaras AFA. 5G Performance Testing of Mobile Chatbot Applications. Lecture Notes in Networks and Systems 2018;p. 1–6. https://ieeexplore.ieee.org/document/8515004.

2. S R, Balakrishnan K. Implementation of an inquisitive chatbot for database supported knowledge bases. Academy Proceedings in Engineering Sciences 2016;41:1173–1178. https://link.springer.com/article/10.1007/s12046-016-0544-1.

3. L, Darima O, Bitner FMJ. Implementation of an inquisitive chatbot for database supported knowledge bases. Academy Proceedings in Engineering Sciences 2019;p. 77–103. https://link.springer.com/chapter/10.1007/978-3-319-98512-1_5.

4. Martinez J, Castillo D. Implementation of an inquisitive chatbot for database supported knowledge bases. International Journal of Language and Linguistics 2015;3:67–76. https://www.sciencepublishinggroup.com/article/10.11648/j.ijll.s.2015030601.19.

5. Ding L, Davies D. The integration of lightweight representation and annotation for collaborative design representation. Research in Engineering Design 2009;20:185–200. https://link.springer.com/article/10.1007/s00163-009-0077-2.

6. Firmansyah, Ilyas M, Kasyidi R, F. Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. Prosiding Industrial Research Workshop and National Seminar 2020;11:488–495. https://link.springer.com/article/10.1007/s00163-009-0077-2.

7. Xueming, Siliang L, Zheng T, Qu FZ. Frontiers: Machines vs humans: The impact of artificial intelligence chatbot disclosure on customer purchases. Science, and undefined 2019 2019;38:937–947. https://pubsonline.informs.org/doi/10.1287/mksc.2019.1192.

8. patrickKeith K. AI in contact centers. Communications of the ACM 2017;60 no. 8:18–19. https://dl.acm.org/doi/10.1145/3105442.

9. Stefanus Christian Susetyo Harjanto ERM. The Relation Among Business Process Orientation Practices in Influencing Organizational Performance. IPTEK The Journal for Technology and Science 2023;34 no. 1:44. https://dl.acm.org/doi/10.1145/3105442.

10. Jane Cleland Huang H, Dumitru C, duan CC, Herrera. Automated support for managing feature requests in open forums. Commun ACM 2009;52 no. 10:68–74. https://dl.acm.org/doi/10.1145/1562764.1562784.

11. Erina, Bayu FN, Bakti K, Dyah AYP, Ary APS, Bagus MS, et al. Performance Analysis of AWS and GCP Cloud Providers. 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom) 2022;https://ieeexplore.ieee.org/document/9865484.

12. Abdullah, Nafis ASM, Saiful TM, Islam. A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP) 2018;p. 1–4. https://ieeexplore.ieee.org/document/8554396.

13. Sharat, Abha S, Navya T, Shivani M, Preeti A, Nagrath. Sentiment Analysis Using Gated Recurrent Neural Networks. SN Computer Science 2020;1 no. 2. https://link.springer.com/article/10.1007/s42979-020-0076-y.

14. Georgios, Heri KP, Helge R, Langseth. Effective hate-speech detection in Twitter data using recurrent neural networks. Applied Intelligence 2018;48:4730–4742. https://link.springer.com/article/10.1007/s10489-018-1242-y.

15. Sudan, Deepak J, Hoang P, David VL, Taniar. Recurrent neural network for detecting malware. Comput Secur 2020;99. https://www.sciencedirect.com/science/article/pii/S0167404820303102?via%3Dihub.

16. Rajat, Pradip KB, Arghya KB, Ray. Cognitive Chatbot for Personalized Contextual Customer Service: Behind the Scene and beyond the Hype. Information Systems Frontiers 2024;26:899–919. https://link.springer.com/article/10.1007/s10796-021-10168-y.

17. Yingzi, Chih X, Patrick HS, van Esch; I, Xu LL. AI customer service: Task complexity, problem-solving ability, and usage intention. Australasian Marketing Journal (AMJ) 2020;28 no. 4:189–199. https://journals.sagepub.com/doi/10.1016/j.ausmj.2020.03.005.

18. Mindi, Jun L, Senchun L, Ruiqi C, Chen C, Baihai Z, et al. A Novel Industrial Intrusion Detection Method based on Threshold-optimized CNN-BiLSTM-Attention using ROC Curve. 2020 39th Chinese Control Conference (CCC) 2020;28 no. 4. https://ieeexplore.ieee.org/document/9188872.

19. Giovanni, Guilherme AS, de Andrade; Geovana G, Francisco RSS, João CMD, Rafael PJDC, et al. A Conversation-Driven Approach for Chatbot Management. IEEE Access 2022;10:8474–8486. https://ieeexplore.ieee.org/document/9681834.

20. Addi, Lili AM, Jiang. KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding over Linked Data. IEEE Access 2020;8:149220–149230. https://ieeexplore.ieee.org/document/9165716.

21. W, Youden J. Index for rating diagnostic tests. Cancer 1950;3 No. 1:32–35. https://acsjournals.onlinelibrary.wiley.com/doi/10.1002/1097-0142(1950)3:1%3C32::AID-CNCR2820030106%3E3.0.CO;2-3.

How to cite this article: Dyah Ayu Permata Sari, Ahmad Saikhu, Ratih Nur Esti Anggraini (2023), HYBRID HUMAN-CONTEXTUAL CHATBOT WITH CONTINUOUS MODEL DEVELOPMENT USING RECURRENT NEURAL NETWORK FOR HELP-DESK SUPPORTING TOOL, *IPTEK The Journal of Technology and Science*, .