



Verbal Question and Answer System for Early Childhood Using Dense Neural Network Method

La Ode Fefli Yarin¹, Zahir Zainuddin¹ and Ingrid Nurtanio¹

¹Departement of Informatics, Hasanuddin University, Makassar, Indonesia

Received 25 March 2024, Revised 29 September 2024, Accepted 3 October 2024

Abstract: Early childhood is a crucial stage for rapid language development, where early childhood children often ask unstructured questions due to their limited ability to formulate questions. Therefore, this study compares the proposed method, namely Dense Neural Network (DNN) with interconnected neuron architecture, with LSTM, BERT, and BART methods to determine which method is more effective in handling questions from early childhood on an Indonesian language corpus consisting of 5,000 questions and answers. The training results show that DNN consistently outperforms LSTM, BERT, and BART in various scenarios. From 200 to 500 epochs, DNN achieved better loss values and a significant increase in accuracy. At 500 epochs, DNN achieved 93.59% accuracy, surpassing BERT 90.51%, BART 88.33%, and LSTM 88.24%. The test with 2000 questions showed an impressive DNN accuracy of 93.1%, compared to LSTM 79.05%, BART 64.3%, and BERT 67.55%. DNN consistently gives correct answers to both structured and unstructured questions, unlike other methods that struggle with unstructured questions. For example, testing the corpus question “Apa itu rusa” with DNN yielded a score of 1, whereas other methods yielded a score of 0 due to differences in structure. These findings highlight the robustness and flexibility of DNN in handling novel or unconventional questions, demonstrating its advantage over LSTM, BART, and BERT in the context of early childhood questions. This research significantly contributes to the development of Natural Language Processing (NLP) systems for early childhood education.

Keywords: Dense Neural Network (DNN), Early childhood, Natural Language Processing (NLP), Unstructured questions, Question answering system.

1. INTRODUCTION

Early childhood is a crucial stage for rapid language development. Children in this age group are known for their inquisitive nature, frequently asking questions. However, their question-forming abilities are still developing, often leading to poorly structured or vague inquiries. Despite this, their questioning behavior plays a crucial role in cognitive development by fostering conversational skills. The development of early childhood speaking skills can be challenging. Technological advancements have brought about innovations in education, aiming to make learning more engaging and interactive. One such technology is Natural Language Processing (NLP), powered by Artificial Intelligence (AI).

NLP is a computational approach that studies how humans utilize, employ, and interpret language [1][2]. AI has enabled significant advancements in computer methods for text comprehension and manipulation. NLP is a cornerstone of AI development, focusing on creating systems that can process human language at or beyond human capabilities.

This includes tasks like speech recognition, text analysis, categorization, translation, and more. [3]. In our research, Question Answering (QA) is a core component of NLP and AI. QA involves answering natural language questions using structured information [4]. QA utilizes various techniques such as text comprehension, information retrieval, and inference to produce accurate and relevant answers [5].

Therefore, this QA system is well-suited for kindergarten learning, as it facilitates early childhood social interaction. A question answering system is a model capable of responding to questions in natural language by retrieving relevant answers from given text [6]. Such systems are crucial today for automating user responses, particularly for young children [7]. Numerous studies emphasize the importance of automated QA systems, making it a prominent research area due to the need for machines to comprehend the context of questions across various domains and languages, including Indonesian [8]. The primary challenge in QA systems is providing the most accurate and precise answers, which necessitates integrating human-like intelligence into



machines [9]. Moreover, these innovative tools can be potentially harmful if misused, so users should be cautious of potential inaccuracies [10].

Educational tools with question-and-answer systems can aid children in expanding their knowledge and developing their communication abilities. Early childhood questions are inputted through speech, and answers are provided through speech. This facilitates the attainment of early childhood education objectives. Research on learning media for early childhood is limited, particularly those that incorporate question-and-answer systems into educational tools. This integration can have a significant impact on child development. A crucial aspect is the question domain for building a question-and-answer system for early childhood, who may not yet be able to formulate questions effectively [11].

A significant challenge in this research is developing a system that can comprehend questions from early childhood children. This is quite complex as their questions often require correct articulation due to limited articulation and poor question-forming abilities. Additionally, the system must be able to interpret the context of the questions posed by children. The developed question-and-answer system aims to provide accurate answers to these questions. Various methods have been employed in Natural Language Processing, particularly in question-and-answer systems. However, challenges include managing non-linear or unordered words and understanding the context of questions [12]. For instance, the corpus question in Indonesian, "apa itu rusa," might be varied by early childhood children into "rusa itu apa," "apa itu hewan rusa," "hewan rusa itu apa," or "apa hewan rusa itu." Consequently, the system may struggle to process and understand these questions effectively, leading to inaccurate and imprecise answers.

2. RELATED WORK

Previous research has applied Recurrent Neural Networks (RNNs) with Long-Short Term Memory (LSTM) mechanisms to develop a question-answering system in the context of early childhood education in Indonesia. The aim of the study was to measure the system's accuracy during testing. The results showed that the RNN model achieved an accuracy of 78.11%, while the LSTM model performed better with an accuracy of 89.5% [7].

The research [12] investigates a question-answering system built with deep learning. The system uses techniques like matching question pairs with a combination of Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) networks. The challenge lies in the complexity and diverse nature of Chinese sentences. Additionally, parsing word meanings and handling specific domains pose difficulties. Despite these challenges, the system achieved an accuracy of 86.38% during testing. CNNs play a role in extracting key features from the text, while BiLSTMs handle in-depth analysis of the meaning for the entire sentence.

Building on this idea, another study proposes a method that leverages big data to train the system. This approach involves constructing a knowledge base from massive amounts of text data. They then combine this knowledge base with a database and Artificial Intelligence Markup Language (AIML) to pinpoint missing information or question formats that the chatbot needs to answer effectively. Furthermore, the study explores the potential of various deep learning techniques for question answering, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRUs) – all adept at handling sequential data like text [13].

Theresa and Mathivanan also explored question-answer systems in their research [14]. Their study proposed a new system that combines three stages: feature extraction, optimization, and learning. First, the system extracts features from text inputs by converting words into vectors. Then, it uses an algorithm called Sail Fish-based Whale Optimization Algorithm (SF-WOA) to choose the most relevant features. Finally, an Adaptive Recurrent Neural Network (A-RNN) tackles the task of learning question-answer relationships. This system is unique because it can continuously learn from new questions. When encountering new questions, SF-WOA dynamically adjusts the importance of existing features, allowing the system to learn beyond its initial training data. In terms of memory efficiency, the proposed SF-WOA-RNN system significantly outperforms other similar approaches (GWO-RNN, WOA-RNN, PSO-RNN, and SFO-RNN) by up to 8.2%.

Previous research on question-answering systems, titled "DAQAS: Deep Arabic Question Answering System based on duplicate question detection and machine reading comprehension," aimed to improve the accuracy of extracting answers from Arabic questions. This system uses deep learning techniques such as BERT, GPT, and T5 to enhance its performance. Various experiments demonstrated the system's effectiveness, achieving an F1 score of approximately 54.7% when retrieving the top 5 relevant segments [15].

Similar research was conducted by Adio and Simeon in 2023 [16]. This study focuses on utilizing product reviews posted online by consumers as a valuable resource while waiting for responses from community question-answer systems. The researchers used the bi-directional auto-regressive transformer (BART) model to classify the ability to answer questions and generate responses. Out of 1000 test questions, 618 were responded to, while 382 remained unanswered.

Research unrelated to question-answer systems but employing Dense Neural Network methods is detailed in the study [17]. This study integrates a machine learning model based on Dense Neural Networks (DNN) into a CFD solver for 2D simulations of laminar premixed methane-air flames undergoing side-wall quenching (SWQ). The

well-established 2D SWQ case encompasses two significant flame regimes: the adiabatic unstretched flame regime and the non-adiabatic flame quenching at the wall. The neural network-based ML chemical model presents a promising alternative to traditional manifold tabulation methods, addressing some limitations. Notably, the DNN model requires only 2% of the QFM memory while maintaining the exact computational cost.

Studies show that question-and-answer systems in English are popular due to their user-friendliness. Understanding what motivates students to use these systems is crucial, as it shapes their overall perception of the benefits. This research takes a student-centred approach, utilizing the PPMH model to analyze how student behaviour towards question-and-answer systems changes. The PPMH framework focuses on four key factors: a student's belief in their ability to succeed (Performance Expectancy), the perceived difficulty of using the system (Effort Expectancy), potential feelings of isolation when using it (Social Isolation), and anxieties related to COVID-19 (COVID-19 Fear). By analyzing student feedback data, the study aimed to identify how these factors influence students' intentions to use the system. The results show that, on average, students scored between 3.72 and 4.10 on Performance Expectancy, Effort Expectancy, and COVID-19 Fear. This suggests these factors have a positive influence (pull effect) on students' willingness to use question-and-answer systems for English learning. While Social Isolation scores ranged from 2.72 to 3.0, indicating a slight push effect (encouraging use despite some concerns), the overall study suggests a positive trend in student behaviour towards using these systems [18].

Research in the healthcare field also employs AI and NLP in question-and-answer systems to enhance communication between virtual healthcare assistants and patients. This assists patients in identifying diseases by inputting questions about their symptoms. The system will remind patients to take their medication based on the medication records provided by the patients to the system, making it easier for patients to learn about their illnesses and motivating them to take preventive measures [19].

A recent study in 2024 [20] explored the evolution of educational chatbots from traditional models to Retrieval Augmented Generation (RAG) techniques. This research aimed to develop a teaching assistant system for data augmentation and text analysis courses at the University of Leeds. The study proposed a RAG-based teaching chatbot that combines retrieval and generative methodologies. The chatbot was implemented using the LangChain framework with the OpenAI GPT-3.5 Turbo model to address limitations such as limited flexibility and difficulty in handling queries outside of predefined intents. The results for correctness were 3.62.

Research on question-and-answer systems has also been conducted using the large language model (LLM) method

in the domains of cardiovascular health, oncology, and psoriasis chatbots. This study tested these chatbots with 45 questions, finding that the shortest answer length was 7.0 (6.0 - 8.0) and the longest was 17.5 (13.0 - 24.5). The study demonstrated that these chatbots, in terms of length, quality, and readability, provide answers based on web-sourced data with high response reliability. This indicates that while people seek information from chatbots, they should exercise caution and verify the answers received, especially when inquiring about medical and health matters [21].

Research on question-and-answer systems in English education indicates that proficiency in English using AI chatbots can accelerate the learning process and help students achieve their course goals. The benefits of this include improving speaking proficiency and reducing anxiety when speaking. However, there is a challenge in this system related to speech recognition faced by the chatbot [22].

The research by Guo and Li [23] involved 69 Chinese undergraduate students participating in workshops on creating chatbots. This study contributes to the literature on chatbot-supported language learning and the use of AI in education. It highlights the potential for students to develop chatbots tailored to their personalized learning needs, which has significant educational implications. By creating chatbots that align with their specific learning goals and preferences, students can enhance their engagement and motivation, resulting in more effective learning outcomes. The study also provides insights into the application of RAG techniques in language classrooms.

Researchers created a chatbot named Babybot specifically designed to answer Pakistani caregivers' questions about childhood immunizations. This chatbot uses natural language processing (NLP) and machine learning (ML) to understand questions asked in the local language and provide answers. To test Babybot's effectiveness, they enrolled caregivers visiting 12 immunization centers and some reached through targeted text messages. The study design combined different methods (mixed-methods approach) to assess how easy it was for caregivers to use Babybot (feasibility) and how much they liked it (acceptance). This included tracking how often caregivers interacted with the chatbot, analyzing data collected by the technology itself, and conducting in-depth interviews with 20 participants. Overall, the study found that Babybot, a local language chatbot using NLP, is a practical and well-received tool for providing real-time information on immunizations in areas with limited resources. This text-based system can potentially reduce the burden on helpline staff and ensure caregivers get the information they need right away, preventing delays or missed vaccinations [24].

The subsequent research serves as a reference for the forthcoming study, although its objectives differ. The current study employs the same method due to its higher accuracy than previous studies. This research uses the

Dense Neural Network (DNN) method to detect cardiac arrhythmia and implements it in a low-cost, power-efficient, low-computation ECG monitoring system. The results show that the developed system can detect arrhythmia with an accuracy of approximately 97.09%. The cardiac arrhythmia detection algorithm using an artificial neural network was applied in this research, and the system effectively detects arrhythmia with a small model size, making it suitable for use on microcontroller units (MCUs) while maintaining high performance [25].

The latest research on question-answer systems focuses on developing a medical chatbot using BERT, the latest deep learning technology, to enhance the communication and accessibility of healthcare services. It addresses traditional challenges faced by medical chatbots, such as misinterpretation of medical conversations, incorrect responses to medical jargon, and lack of personalized feedback. By leveraging Bidirectional Encoder Representations from Transformers (BERT), the chatbot effectively overcomes these issues. The performance metrics of the chatbot are impressive, with an accuracy of 98%, ensuring high precision in handling medical inquiries. Additionally, a precision score of 97% underscores the chatbot's accuracy and reliability in providing medical responses [26].

Previous research on question-answering systems, conducted by Utomo, employed Long Short-Term Memory (LSTM) on a dataset of early childhood questions, achieving an accuracy of 89.5%. While LSTM excels at processing sequential words and maintaining context in questions requiring chronological order, it struggles with more complex contexts due to its unidirectional processing. Other studies using Bidirectional Encoder Representations from Transformers (BERT) for medical question answering reported an accuracy of 98% with a precision of 97%. BERT demonstrated significant advantages in understanding context from both directions, allowing the model to capture more complete word relationships. However, BERT demands higher computational resources and longer training time compared to other methods, which can be a hindrance for real-time applications or systems with limited resources.

Bidirectional and Auto-Regressive Transformers (BART), when tested on 1000 questions, successfully answered 618 while failing to answer 382. BART combines an encoder-decoder architecture with BERT-like capabilities but struggles with highly specific questions or when relevant training data is limited. Dense Neural Networks (DNN), commonly used outside of question-answering systems, have shown an accuracy of 97.09% due to their fully connected inputs, allowing for richer feature processing. However, DNNs are susceptible to noise from irrelevant inputs, especially when questions contain words unrelated to the main topic. Therefore, this study will compare the proposed Dense Neural Network (DNN) method with LSTM, BERT, and BART in handling questions from young children. Children's questions

often require deeper understanding due to their limited articulation and ability to formulate questions clearly. While DNN offers advantages in processing interconnected features, this research aims to evaluate whether the DNN approach can overcome its primary weakness sensitivity to noise from irrelevant inputs compared to more advanced methods like BERT and BART.

3. METHODOLOGY

This research commences with the development of a PAUD corpus dataset comprising approximately 5000 Indonesian questions. The dataset was compiled from various sources, including direct interactions with preschool children, educational reference books, and online resources. The uniqueness of this dataset lies in its question-answer pair structure, which will be used to train an artificial intelligence model. The trained AI model will then be implemented on two different devices with specifications as detailed in Table I.

TABLE I. Device specifications

Device	A	B
OS	Windows 10 Pro	Windows 10 Pro
Processor	11th Gen Intel®	Intel®
Frequency	2.50 GHz	1.70 GHz
RAM	32 GB	6 GB
Core	i9-1190F	i3-4005U

In Table I, Device A is designated as the training device, while Device B will serve as the testing device. The overall system implementation is visualized in Figure 1. Both Device A and Device B will be installed with the Python environment and appropriate supporting software. Complete configuration specifications for both devices can be found in Table II.

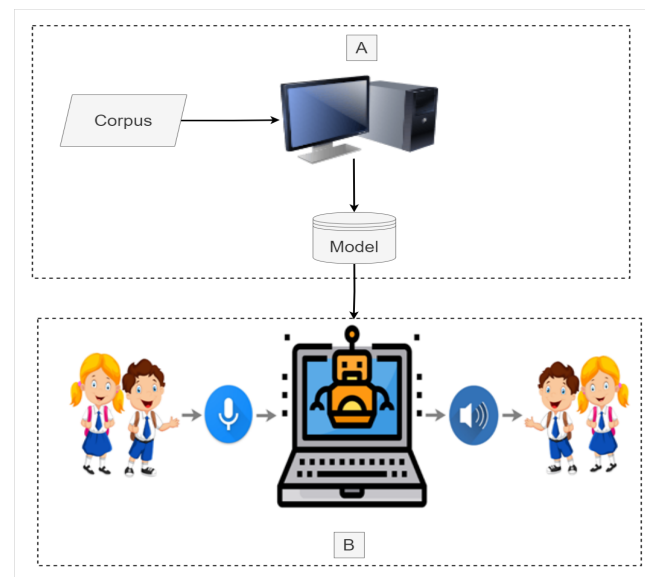


Figure 1. Block Diagram of the Scenario to be Performed

TABLE II. Table of Categories and their Corresponding Tools.

Category	Tools
Speech Recognition	speech_recognition
Text-to-Speech	gTTS
NLP Tokenizing	nlk
Neural Network Layer	Tensorflow.keras.layer
Optimization	Tensorflow.keras.optimizer

The tools listed in Table II will be employed for deep learning tasks such as speech recognition, natural language processing, and the construction of dense neural networks, incorporating activation functions, dropout, and optimizers. This research will involve two primary stages: the Preprocessing Stage and the Training Phase.

The model development process involves several main stages. One important stage is word tokenization, where questions and answers are broken down into smaller units called "tokens". The next stage is embedding, which involves transforming these tokens into numerical vector representations that represent semantic meaning in a multidimensional space. The results of tokenization and embedding are then used in the proposed method, namely the Dense Neural Network (DNN). The DNN model is trained to understand and provide highly accurate answers. The DNN training process is done in a gradual manner, where the model continuously adjusts its internal parameters to improve its predictive ability and understanding of human language. Further details regarding the testing process are explained in Figure 2.

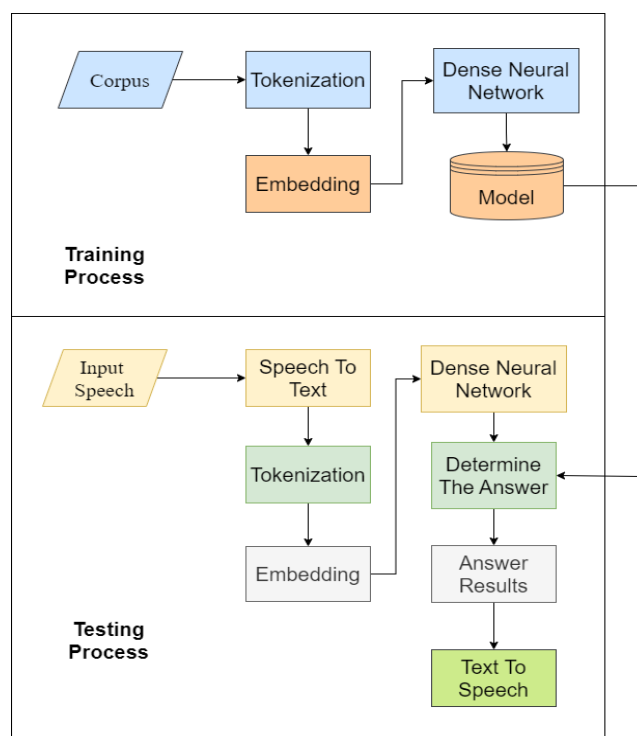


Figure 2. Blok diagram of the Question and Answer System

In the proposed system, early childhood children will pose various questions categorized into three types: structured, semi-structured, and unstructured, while maintaining a similar context. These questions will be classified using the Dense Neural Network (DNN) method to find matches with questions in the corpus. Additionally, by conducting various testing scenarios, this research can provide a comprehensive overview of how well the system can adapt and perform in different situations. This research aims to evaluate the system's ability to overcome existing challenges. This thorough evaluation is crucial for identifying the system's strengths and weaknesses, which will guide further development and improvement. Figure 3 illustrates the testing scenarios to be conducted.

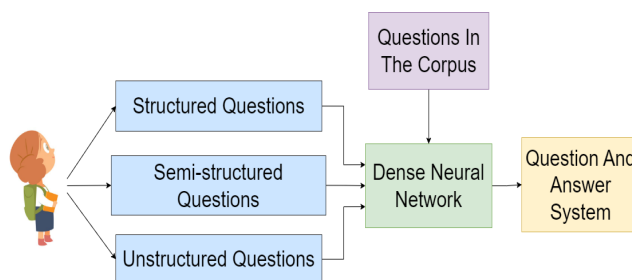


Figure 3. Block Diagram of the Scenario to be Performed

A. Preprocessing Stage

1) Corpus

A corpus is a collection of text datasets used for linguistic analysis or language technology development. Contemporary corpus construction methods strive to maximize coverage by gathering data from diverse sources [27]. This study focuses on creating an Indonesian language corpus specifically tailored to early childhood questions and answers. The corpus comprises 5000 question-answer pairs, meticulously collected and organized to form a comprehensive and varied database. Table III presents the types of early childhood question datasets included.

TABLE III. Number of questions for each question type

Question	Count	Min Words	Max Words
Conversation	470	1	14
Self Introduction	156	2	4
Fruits	576	3	13
Counting	490	1	6
Alphabet	209	3	7
Animals	785	3	14
Islamic Religion	409	2	7
Object Shape	450	3	5
Flowers	550	4	6
Colors	560	3	5
Profession	345	3	5

Table III presents eleven question types, with a minimum of one word and a maximum of fourteen words per question. An example dataset of conversation-type questions is shown in Table IV.

TABLE IV. Sample corpus of questions and answers in Indonesian

Question	Answer
Hai teman	Hai juga teman
Selamat pagi	Selamat pagi juga
Selamat malam	Selamat malam juga
Siapa nama kamu?	Aku tidak punya nama
Kenapa kamu tidak punya nama?	Karena aku robot

Table IV shows an example of the corpus data that we use in the table, and there are two columns: the first column is the question column, and the second is the answer column. The question column consists of five questions in Indonesian, namely "Hai teman," "Selamat Pagi," "Selamat malam," "Siapa nama kamu?" and the last, "Kenapa kamu tidak punya nama?" in English, the questions are "Hi friend," "Good morning," "Good night," "What is your name?" and the last "why don't you have a name." The answer column consists of five answers in Indonesian, namely "Hai juga teman," "Selamat pagi juga," "Selamat malam juga," "Aku tidak punya nama," and the last, "Karena aku robot" in English "Hi there, friend," "Good morning too," "Good night too," "I do not have a name," and the last "Because I am a robot." The system used will read from the data above before processing at the next stage, which is the tokenization process.

2) Tokenization

The tokenization process is the process of breaking sentences or documents into pieces of words [28]. This process is essential in (NLP) tasks, transforming text data into manageable units for further analysis and processing. By breaking down sentences or documents into individual tokens, generally words or subwords, tokenization facilitates tasks such as text classification, sentiment analysis, and machine translation. This process allows algorithms to understand the text's structure and meaning better, resulting in more accurate and efficient NLP models. Tokenization is carried out in this study so that the pieces of words can be embedded in each word more details in Figure 4.

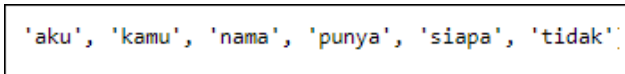


Figure 4. Example of Indonesian Tokenization Results

The tokenization result shown in Figure 4 is the question "siapa nama kamu?" in English, "What is your name?" and the answer "aku tidak punya nama" in English, "I do not have a name," the tokenization results of these questions, for example, the word "siapa," is cut and separated from the word "nama kamu" first after this the word "nama" is separated again from the word "kamu" so that the question is cut into "siapa," "nama," "kamu" while in the answer "aku tidak punya nama" if tokenized then the word "aku" will be cut first from the word "tidak punya nama," after that the word "tidak" is cut or separated from the word "punya nama," after the word not then the next word "punya" is

separated from the word "nama" so that the answer already consists of four words that have been separated, namely "aku," "tidak," "punya," "nama." However, when the word appears, it will not display the repeated word and will remove symbols such as question marks.

3) Embedding

At this stage, the process of converting words into vector representations in multidimensional space is called word embedding [29]. Word embedding is used in this study to adapt the initial model with light supervision online to improve the performance of the spoken language understanding module [30]. Word embedding techniques make it possible to represent words in vectors that preserve the semantic relations between words in the text, thus enabling more accurate and efficient modeling in natural language processing. In addition, using word embedding, the model can recognize and capture complex meaning relationships between words, enabling deeper interpretation of the processed text data. Figure 5 shows the results of embedding the question-and-answer words into vectors.

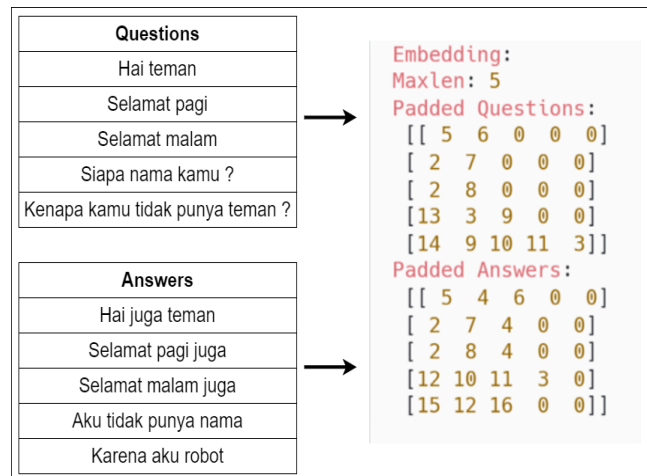


Figure 5. Embedding Results of Questions and Answers

Figure 5 shows an example of the embedding results of questions and answers; in the embedding results, for example, "Hai teman," the question has a vector value [5 6 0 0 0] and "Kenapa kamu tidak punya nama," the question has a vector value [14 9 10 11 3]. In vector [5 6 0 0 0], after vector values 5 and 6, the value displayed is 0 3 times because the word in the question "Hai teman" has only two words. In comparison, 0 3 times is an empty value and is the max word length of the question in the corpus, namely vector [14 9 10 11 3] with many words in the question of five words.

B. Training Phase

1) Dense Neural Network (DNN)

A Dense Neural Network (DNN) is the simplest form of artificial neural network. This computational model is biologically inspired and designed to mimic how the human

brain processes information [31], [32]. DNN consists of several hidden layers, each comprising a set number of fully connected or "dense" neurons. Each layer contains a predetermined number of neurons connected from one layer to the next as inputs [33]. DNN can analyze complex data patterns and has been applied in various disciplines for various applications, including regression analysis, classification, and unsupervised data clustering. The structure of DNN consists of three main parts: the input layer, hidden layers, and the output layer. The input layer of a DNN is the first layer of the network that receives numerical data or vectors. The hidden layers can consist of one or more between the input and output layers. The output layer is the layer that provides the final result of the network. DNNs also employ activation functions, optimization techniques, and regulation techniques. The Dense Neural Network (DNN) architecture is shown in Figure 6.

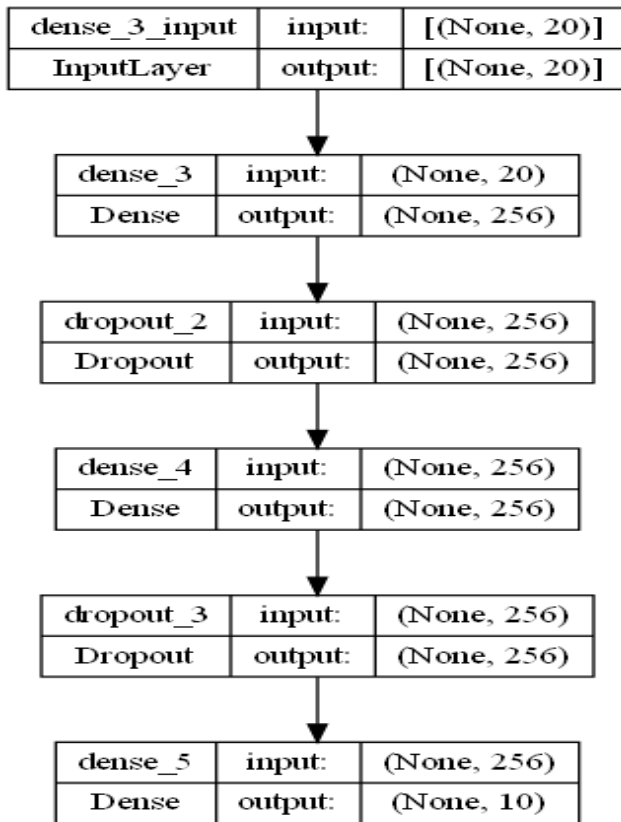


Figure 6. Dense Neural Network (DNN) Architecture

The compact mathematical equation form of a Dense Neural Network is as follows.

$$z = \sigma(W \cdot x) + b \quad (1)$$

Where, z = The resulting weighted value before applying the activation function, σ = Activation function used in the hidden layer, W = Weight matrix for the hidden layer, x = Network input, b = Bias vector for the hidden layer.

a) Activation Functions

In this study, two activation functions are utilized: the ReLU activation function and the Softmax function. ReLU is employed to achieve faster convergence during training and to mitigate the vanishing gradient issue [34]. The formula for the ReLU function applied in the first and second hidden layers of the dense neural network is as follows:

$$f(z) = \max(0, z) \quad (2)$$

Where, $f(z)$ = Is the activation function itself, z = Input received by the neuron, $\max(0, z)$ = Maximum value between 0 and z .

In the ReLU function, input values that are negative or zero are converted to zero, while values greater than zero are retained. This enables the neural network to learn a non-linear representation of the input data, which is important for capturing complex and abstract features in the data.

The dense neural network method in this study also utilizes activation functions, specifically Softmax, for the output classification task. These activation functions are required to convert input values into interpretable outputs, with Softmax being used in particular to generate probability distributions of possible classes. The following Softmax formula is given by [25]. By using Softmax, the model can provide a prediction of the most class for each instance of input data, and it is important to understand how a DNN translates its output into a classification task.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}} \quad (3)$$

Where, $\sigma(z_i)$ = Output Softmax for class k , k = Total number of classes in the classification task, z = Input vector to softmax function, (z_i) = k -th element of vector z , e^z = Exponential of the logit (z_k) , $\sum_{k=1}^K e^{z_k}$ = Sum of the exponentials of all logits in the vector z .

b) Regulation Technique

In addition to using optimizers and activation functions, this study also applies the dropout regulation technique. This method is employed to address overfitting by reducing neuron co-adaptation during training. The technique works by deactivating or dropping out specific neurons in the layer during exercise, making the system more robust and less dependent on particular neurons. The dropout rate is 0.5 or 50%, a mid-range value to ensure a balance between deactivated and retained neurons.

c) Optimization

A Stochastic Gradient Descent (SGD) optimization is used to optimize the method in this study. SGD is one of the most fundamental and commonly used optimization algorithms in machine learning, particularly for training neural networks [34]. SGD is an advanced optimization approach that handles large datasets effectively [35]. Here is an explanation of the parameters mentioned in your question and why SGD is used.

- `tf.keras.optimizers.legacy.SGD`: This function is used to create an instance of the SGD (Stochastic Gradient Descent) optimizer from the Keras library.
- `learning_rate=0.01`: The learning rate determines the step size taken to update the weights at each iteration. The learning rate used is 0.01.
- `decay=1e-6`: This parameter gradually decreases the learning rate after each update. This helps achieve more stable convergence.
- `momentum=0.9`: Momentum helps accelerate convergence and reduce oscillations by maintaining the previous gradients.
- `nesterov=True`: Enables Nesterov Accelerated Gradient, which considers the future position of the gradient for more accurate and responsive updates.

d) Model performance evaluation

Accurate and loss metrics are used to evaluate the model's training results. The formulas for accuracy and loss are as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Where, TP = (True Positive), Which is the number of predictions that are true positive, FP = (False Positive), Which is the number of false positive predictions, TN = (True Negative), Which is the number of correct negative predictions, FN = (False Negative), Which is the number of false negative predictions.

$$Loss = - \sum_{i=1}^N y_i \log(p_i) \quad (5)$$

where, y_i = the true label for class i , p_i = the predicted probability for class i , and N = the number of classes.

For more details, see the pseudocode showing the model flow created in Figure 7.

During the training process of a Dense Neural Network, the initial layers will receive vector values obtained from the embedding process, where the inputs are based on the number of words in the corpus, represented as $(x_1, x_2, x_3, \dots, x_n)$. These values will be passed to each neuron, the number of which is determined based on requirements. Mathematical operations will then be performed as $z = \sigma(W \cdot x) + b$, where W is the weight matrix obtained from the existing network, x is the input from the previous layer, and b is the bias of each neuron in the hidden layer. The hidden layer can consist of multiple layers depending on the needs. After performing the mathematical operation $\sigma(W \cdot x) + b$, the ReLU activation function will be applied to address values less than zero, such as negative values, and if the values are greater than zero, the initial values will be retained. During training, dropout will also be applied after the

```
# Initialize training process for Dense Neural Network

# Retrieve vector values from the embedding process based on the number of words in the corpus
INPUTS = (x_1, x_2, x_3, ..., x_n)

# Process inputs through the first dense layer
# Dense Layer 1: 256 neurons
z = W_DENSE_1 * INPUTS + b_DENSE_1
ACTIVATED_OUTPUT_DENSE_1 = ReLU(z)

# Apply Dropout with a probability of 0.5
DROPOUT_OUTPUT_1 = APPLY_DROPOUT(ACTIVATED_OUTPUT_DENSE_1, probability=0.5)

# Process inputs through the second dense layer
# Dense Layer 2: 256 neurons
z = W_DENSE_2 * DROPOUT_OUTPUT_1 + b_DENSE_2
ACTIVATED_OUTPUT_DENSE_2 = ReLU(z)

# Apply Dropout with a probability of 0.5
DROPOUT_OUTPUT_2 = APPLY_DROPOUT(ACTIVATED_OUTPUT_DENSE_2, probability=0.5)

# Process inputs through the final dense layer
# Dense Layer 3: 10 neurons (Output layer)
z = W_DENSE_3 * DROPOUT_OUTPUT_2 + b_DENSE_3
FINAL_OUTPUT = Softmax(z)

# If the loss value is still high, perform backpropagation
IF LOSS_VALUE is HIGH:
    # Update weights and biases using SGD optimization
    UPDATE_WEIGHTS_AND_BIASES_USING_SGD()

# End of process
```

Figure 7. Pseudocode of DNN

hidden layer or the selected layer based on requirements, with a probability of 0.5, meaning that 50% of certain neurons will be randomly deactivated to prevent overfitting. If there are more hidden layers, the process will be repeated as before. However, if there are no more hidden layers, the output from the last layer will be fed into the Softmax activation function for classification, converting the input values into probabilities that sum up to 1. If the loss value remains high, backward propagation will be conducted to update the weights and biases using SGD optimization.

4. RESULT AND DISCUSSION

In the results and discussion section, we will convert the questions and answers from Text format to JSON format for the corpus dataset. This conversion is necessary because previous research [7] used the text format to structure questions and answers. However, to achieve variation in questions and answers, multiple instances of the same meaning need to be created, which is inefficient. Therefore, this study will use a different format, namely JSON. This choice is based on research [36] that demonstrates JSON's ability to accommodate various patterns of questions and

answers within a single question tag, thereby enhancing efficiency and diversity in representation. For example, we will take several questions and answers in Indonesian, such as "Selamat pagi," which translates to "Good morning" in English. The answers can have several patterns, such as "Selamat pagi," "Pagi juga," and "Pagi." In English, these correspond to "Good morning," "Good morning to you," and "Morning." Figure 8 shows the result of converting the text corpus dataset to JSON.

Question	Answer
hai teman	hai juga teman
selamat pagi	selamat pagi juga
selamat malam	selamat malam juga
siapa nama kamu	aku tidak punya nama
kenapa kamu tidak punya nama ?	karena aku robot

↓

```

{
  "intents": [
    {
      "tag": "question_1",
      "patterns": ["hai teman", "hai"],
      "responses": ["hai juga teman"]
    },
    {
      "tag": "question_2",
      "patterns": ["selamat pagi"],
      "responses": ["selamat pagi juga", "pagi juga", "pagi"]
    },
    {
      "tag": "question_3",
      "patterns": ["selamat malam"],
      "responses": ["selamat malam juga"]
    },
    {
      "tag": "question_4",
      "patterns": ["siapa nama kamu"],
      "responses": ["aku tidak punya nama"]
    },
    {
      "tag": "question_5",
      "patterns": ["kenapa kamu tidak punya nama ?"],
      "responses": ["karena aku robot"]
    }
  ],
}

```

Figure 8. Text Corpus Dataset Conversion to JSON

During the training process, the 5000-question-and-answer corpus dataset will be configured with hyperparameters used for the LSTM, DNN, BERT, and BART algorithms. These hyperparameters will cover a range of epoch values from 100 to 500. Additionally, the training process will set the batch size to a fixed value of 16 and adjust other crucial settings to optimize model performance. More detailed information on these settings can be found in Table V for deeper understanding.

Table V provides a comprehensive overview of the training process scenario. The optimizer used for the input layers of LSTM, BERT, and BART in this study is Adaptive Moment Estimation (Adam), as this choice is suitable given the large dataset and the stability required for these methods. In comparison, the optimizer for DNN uses Stochastic Gradient Descent (SGD) because it provides better generalization. The learning rate for both Adam and

TABLE V. Hyper Parameter Settings

Hyper parameter		
Input Layer	Learning Rate	0.01
	Batchsize	16
	Epoch	100 - 500
LSTM	Optimizer	Adam
	LSTM Node	256
	LSTM 1 Activation	Relu
	LSTM layer	2
	Dropout	0.5
DNN	LSTM 2 Activation	Softmax
	Optimizer	SGD
	DNN Node	256
	DNN 1 Activation	Relu
	DNN layer	2
BERT	Dropout	0.5
	DNN 2 Activation	Softmax
	Optimizer	Adam
	BERT Node	256
	BERT 1 Activation	Relu
BART	BERT layer	12
	Dropout	0.5
	BERT 2 Activation	Softmax
	Optimizer	Adam
	BART Node	256
BART	BART 1 Activation	Relu
	BART layer	12
	Dropout	0.5
	BART 2 Activation	Softmax

SGD optimizers is set at 0.01, with a consistent batch size of 16 across the varying epochs, ranging from 100 to 500. In all methods, each layer is configured with 256 nodes, using the Rectified Linear Unit (ReLU) activation function for DNN, as ReLU deactivates neurons for negative inputs (producing zero output), resulting in sparsity in activation, which can enhance computational efficiency. Meanwhile, LSTM, BERT, and BART use the GELU activation function because it provides a smoother activation compared to ReLU by combining the properties of the sigmoid function, resulting in a softer effect. To address overfitting and underfitting issues, a dropout rate of 0.5, considered an optimal middle ground, is applied [37], [38]. Additionally, the softmax activation function is used to classify the output results produced by the model.

During the training tests using the DNN method and the comparative method with the hyperparameters listed in Table V, the initial tests will be conducted starting with the lowest number of epochs, which is 100, up to the highest number of epochs in this study, which is 500. The goal is to observe the increase in accuracy and the decrease in loss achieved, thus highlighting the advantages of each method, especially the proposed method with the applied hyperparameters. The results of each method can be seen in Table VI, which shows the training outcomes that have been conducted.

TABLE VI. Comparison of Accuracy Results from LSTM and DNN Training

Metode	Epoch	Accuracy	Loss
LSTM	100	0.4792	1.4006
	200	0.4304	1.4304
	300	0.6987	0.7227
	400	0.7385	0.6626
	500	0.8824	0.3820
DNN	100	0.5048	1.9855
	200	0.8700	0.4440
	300	0.8946	0.3177
	400	0.9183	0.2321
	500	0.9356	0.1809
BERT	100	0.0474	4.9877
	200	0.6149	1.0613
	300	0.5034	1.1958
	400	0.8542	0.3861
	500	0.9051	0.2649
BART	100	0.3141	2.4897
	200	0.6588	0.9062
	300	0.7563	0.6033
	400	0.8542	0.3545
	500	0.8542	0.3048

In Table VI, the experimental findings from the training results highlight the superior performance of the proposed DNN method compared to the LSTM [39], BERT, and BART methods. Across various scenarios, DNN consistently outperforms the other three methods. For instance, in the initial scenario with an epoch value of 100, DNN shows good performance with an initial accuracy of 0.5048. Although the loss value is higher than that of LSTM, it is better than the losses of BERT and BART. However, from epoch 200 to 500, DNN achieves better loss values than the other methods, with a loss of 0.4440 at epoch 200, 0.3177 at epoch 300, 0.2321 at epoch 400, and 0.1809 at epoch 500. DNN also shows improved accuracy from epoch 200 to 500, outperforming the other methods. For example, at epoch 200, DNN achieves an accuracy of 0.8700, followed by BART at 0.6588, BERT at 0.6149, and LSTM, which decreases in accuracy to 0.4304. At epoch 300, DNN achieves an accuracy of 0.8946, followed by BART at 0.7563, LSTM increasing to 0.6987, and BERT at 0.5034. At epoch 400, DNN maintains its performance with an accuracy of 0.9183, followed by BERT and BART, both at 0.8542, and LSTM at 0.7385. In the final epoch, epoch 500, DNN continues to improve its performance with an accuracy of 0.9359, followed by BERT at 0.9051, while BART and LSTM have nearly identical accuracies of 0.8833 and 0.8824, respectively. To better understand the comparison of accuracy and loss values, Figures 9 and Figures 10 present graphical representations illustrating the performance differences between DNN and three comparative methods across various conducted training scenarios.

After the training, the testing phase will be conducted in an early childhood environment. During this stage, children

Comparison of the accuracy of DNN, LSTM, BERT, and BART methods

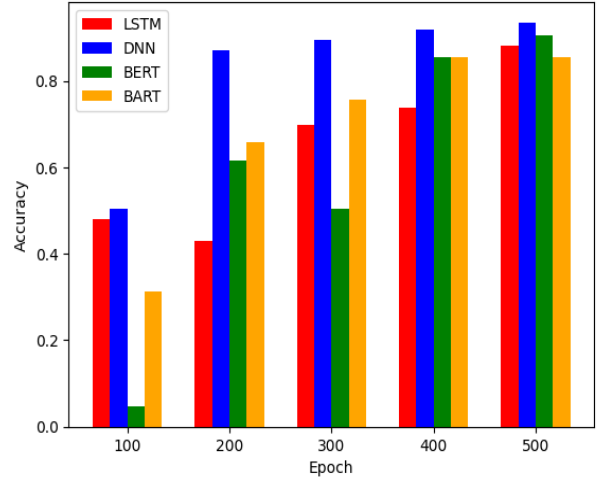


Figure 9. Comparison Chart of DNN, LSTM, BERT, and BART Accuracy

Comparison of the loss of DNN, LSTM, BERT, and BART methods

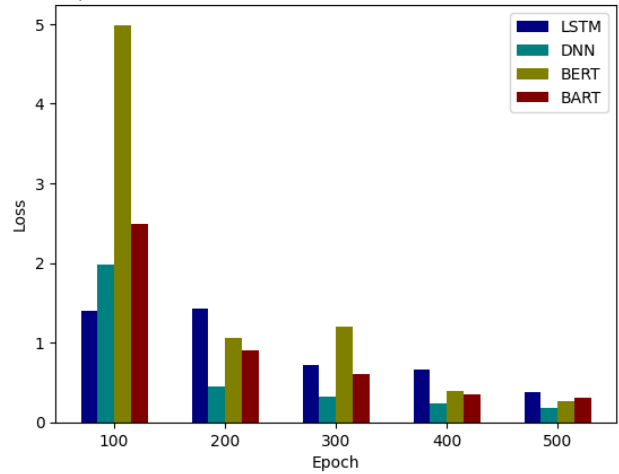


Figure 10. Comparison Chart of DNN, LSTM, BERT, and BART Loss

will interact with the system by asking questions verbally. The system will then process these questions to predict answers using the developed model and respond by verbally articulating the answers. The testing results are shown in Table VII, which consists of questions from the created corpus. Two questions from the corpus were tested with several different early childhood children, resulting in ten testing questions, with five testing questions for each corpus question. Each method is given a correct (1) or incorrect (0) status based on the responses. The experimental results are also shown in Figure 11.

Table VII shows the system test results, where the corpus question "berapa banyak kaki di seekor bebek" is tested with the test question "berapa kaki bebek" across various methods. The DNN method scored 1, while other methods scored 0. This is due to the different structures of

```
berapa kaki bebek
1/1 [=====] - 0s 44ms/step
Sistem: seekor bebek memiliki 2 kaki.

-----

berapa banyak kaki bebek
1/1 [=====] - 0s 35ms/step
Sistem: seekor bebek memiliki 2 kaki.

-----

berapa banyak kaki di seekor bebek
1/1 [=====] - 0s 46ms/step
Sistem: seekor bebek memiliki 2 kaki.

-----

ada berapa kaki di seekor bebek
1/1 [=====] - 0s 47ms/step
Sistem: seekor bebek memiliki 2 kaki.

-----

bebek memiliki berapa kaki
1/1 [=====] - 0s 47ms/step
Sistem: pelikan bisa hidup hingga 15-25 tahun tergantung pada spesiesnya.

apa itu rusa
1/1 [=====] - 0s 31ms/step
Sistem: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup
p di hutan atau daerah yang memiliki banyak tanaman

-----

rusa itu apa
1/1 [=====] - 0s 40ms/step
Sistem: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup
p di hutan atau daerah yang memiliki banyak tanaman

-----

apa itu hewan rusa
1/1 [=====] - 0s 36ms/step
Sistem: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup
p di hutan atau daerah yang memiliki banyak tanaman

-----

hewan rusa itu apa
1/1 [=====] - 0s 31ms/step
Sistem: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup
p di hutan atau daerah yang memiliki banyak tanaman

-----

apa hewan rusa itu
1/1 [=====] - 0s 39ms/step
Sistem: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup
p di hutan atau daerah yang memiliki banyak tanaman
```

(a) DNN

```
You: berapa kaki bebek
1/1 [=====] - 0s 33ms/step
system: Mereka disebut "quailfish" karena bentuk tubuh dan penampilannya yang menyerupai burung
quail (burung puyuh) di darat.
You: berapa banyak kaki bebek
1/1 [=====] - 0s 24ms/step
system: Good bye!
You: berapa banyak kaki di seekor bebek
1/1 [=====] - 0s 25ms/step
system: alpaka disukai karena bulunya yang lembut dan wajahnya yang lucu
You: ada berapa kaki di seekor bebek
1/1 [=====] - 0s 29ms/step
system: Tentu! Lima huruf vokal adalah A, E, I, O, dan U.
You: bebek memiliki berapa kaki
1/1 [=====] - 0s 27ms/step
system: bunglon punya tubuh kecil dengan kulit bersisik dan mata yang bisa bergerak secara ind
ependen. mereka punya lidah yang panjang untuk menangkap mangsa
You: apa itu rusa
1/1 [=====] - 0s 31ms/step
system: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup di hutan atau
daerah yang memiliki banyak tanaman
You: rusa itu apa
1/1 [=====] - 0s 24ms/step
system: hamster suka makan biji-bijian, sayuran kecil, dan buah-buahan.
You: apa itu hewan rusa
1/1 [=====] - 0s 28ms/step
system: Panther adalah jenis hewan kucing besar yang memiliki bulu berwarna hitam dengan bercak
-bercak kecil yang tidak terlihat dengan jelas.
You: hewan rusa itu apa
1/1 [=====] - 0s 30ms/step
system: hiu suka memakan ikan, hewan laut kecil, dan kadang-kadang mamalia laut
You: apa hewan rusa itu
1/1 [=====] - 0s 19ms/step
system: Makanan utama mamot adalah tumbuhan, seperti rumput, akar, dan tunas. Mereka juga mema
kan serangga, telur burung, dan kadang-kadang bangkai kecil
```

(c) BART

```
berapa kaki bebek
1/1 [=====] - 0s 37ms/step
sistem: 26

berapa banyak kaki bebek
1/1 [=====] - 0s 29ms/step
sistem: seekor bebek memiliki 2 kaki.

berapa banyak kaki di seekor bebek
1/1 [=====] - 0s 24ms/step
sistem: seekor bebek memiliki 2 kaki.

ada berapa kaki di seekor bebek
1/1 [=====] - 0s 23ms/step
sistem: seekor bebek memiliki 2 kaki.

bebek memiliki berapa kaki
1/1 [=====] - 0s 25ms/step
sistem: ya, bebek bisa terbang. meskipun mereka lebih nyaman berenang, m
ereka juga bisa terbang untuk mencari makanan atau tempat berlindung

apa itu rusa
1/1 [=====] - 0s 23ms/step
sistem: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung
ng hidup di hutan atau daerah yang memiliki banyak tanaman

rusa itu apa
1/1 [=====] - 0s 33ms/step
sistem: wartawan

apa itu hewan rusa
1/1 [=====] - 0s 38ms/step
sistem: Panther adalah jenis hewan kucing besar yang memiliki bulu berwa
rna hitam dengan bercak-bercak kecil yang tidak terlihat dengan jelas.

hewan rusa itu apa
1/1 [=====] - 0s 23ms/step
sistem: ya, hewan dara bisa terbang dan mereka sering terlihat terbang d
i langit kota.

apa hewan rusa itu
1/1 [=====] - 0s 34ms/step
sistem: ya, hewan dara bisa terbang dan mereka sering terlihat terbang d
i langit kota.
```

(b) LSTM

```
You: berapa kaki bebek
1/1 [=====] - 0s 18ms/step
system: kelelawar berkembang biak dengan cara yang mirip dengan mamalia lainnya. Mereka memiliki
anak dalam bentuk bayi dan merawat mereka sampai cukup mandiri
You: berapa banyak kaki bebek
1/1 [=====] - 0s 20ms/step
system: nyamuk memiliki mata yang bisa mendeteksi cahaya, tetapi penglihatan mereka tidak begitu
baik.
You: berapa banyak kaki di seekor bebek
1/1 [=====] - 0s 18ms/step
system: Melindungi habitat rusa penting karena hal ini membantu menjaga keseimbangan alam dan kea
nekaragaman hayati di hutan dan daerah lain di mana mereka tinggal
You: ada berapa kaki di seekor bebek
1/1 [=====] - 0s 19ms/step
system: Ya, ikan marlin sangat suka berenang cepat. Mereka adalah salah satu ikan tercepat di lau
tan
You: bebek memiliki berapa kaki
1/1 [=====] - 0s 19ms/step
system: kelelawar berkembang biak dengan cara yang mirip dengan mamalia lainnya. Mereka memiliki
anak dalam bentuk bayi dan merawat mereka sampai cukup mandiri
You: apa itu rusa
1/1 [=====] - 0s 18ms/step
system: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup di hutan atau da
erah yang memiliki banyak tanaman
You: rusa itu apa
1/1 [=====] - 0s 19ms/step
system: Rusa adalah hewan mamalia yang memiliki kaki panjang dan cenderung hidup di hutan atau da
erah yang memiliki banyak tanaman
You: apa itu hewan rusa
1/1 [=====] - 0s 19ms/step
system: Dugong adalah mamalia laut yang besar, juga dikenal sebagai 'sapi laut' atau 'paus berlum
ba-lumba.' Mereka memiliki tubuh yang besar dan lonjong, ekor mirip paus, dan hidung yang menyeru
pai gajah.
You: hewan rusa itu apa
1/1 [=====] - 0s 19ms/step
system: rajawali membuat sarangnya di tempat yang tinggi dan aman, seperti di pohon atau tebing.
Mereka menggunakan ranting dan bahan-bahan lainnya
You: apa hewan rusa itu
1/1 [=====] - 0s 18ms/step
system: Hari Ibu penting karena ibu adalah sosok yang selalu menyayangi, merawat, dan mendukung k
ita dengan penuh kasih sayang.
```

(d) BERT

Figure 11. Testing results from (a) DNN, (b)LSTM, (c) BART, and (d) BERT methods



TABLE VII. Example of Comparative Test Results between Methods Applied to a Question-and-Answer System Aimed at early childhood children

Question corpus	Question testing	DNN	LSTM	BART	BERT
Berapa banyak kaki di seekor bebek	Berapa kaki bebek	1	0	0	0
Berapa banyak kaki di seekor bebek	Berapa banyak kaki bebek	1	1	0	0
Berapa banyak kaki di seekor bebek	Berapa banyak kaki di seekor bebek	1	1	0	0
Berapa banyak kaki di seekor bebek	Ada berapa kaki di seekor bebek	1	1	0	0
Berapa banyak kaki di seekor bebek	Bebek memiliki berapa kaki	0	0	0	0
Apa itu rusa	Apa itu rusa	1	1	1	1
Apa itu rusa	Rusa itu apa	1	0	0	1
Apa itu rusa	Apa itu hewan rusa	1	0	0	0
Apa itu rusa	Hewan rusa itu apa	1	0	0	0
Apa itu rusa	Apa hewan rusa itu	1	0	0	0

the test questions, as they lack the words "banyak" and "di seekor," resulting in a structure that differs from the corpus dataset. However, this is not an issue for DNN because this method predicts questions based on relevant words from the test question and the corpus question, so if a word is missing in the test question, the existing words will be processed. This differs from other methods, such as LSTM, which scored 0 because the test question's word order does not match the corpus question's word order, resulting in a different question pattern. In this case, LSTM might still provide the correct answer if the missing word is not in the middle of the relevant words, such as "kaki" between "banyak" and "di seekor." The BART method scored 0 because the test question pattern differs from the corpus question pattern. Meanwhile, the BERT method scored 0 because the given test question does not match the existing context, as this method can understand the context if trained with sufficient data.

In the second testing, the question "berapa banyak kaki bebek." The DNN method scored one because, similar to the previous case, only the phrase "di seekor" is missing, but the test question contains words relevant to the corpus question, which is not an issue for the DNN method. The LSTM method also scored one because the missing phrase "di seekor" still allows the original structure to be retained, enabling the system to make the correct prediction. However, the BART and BERT methods could not provide the correct answer, possibly due to the complexity of the context and a lack of training datasets. The test question is identical to the corpus question in the third testing. The DNN method scored one because the testing and corpus questions are the same, and the LSTM method scored one for the same reason as DNN. However, the BART and BERT methods still scored zero. The fourth testing used a slightly different question, with the addition of the word "ada" in the test question "ada berapa kaki di seekor bebek." The results show that both DNN and LSTM scored one, but the DNN method could have scored zero if the weights and biases provided during training were inaccurate.

For the LSTM method, it can ignore the word "ada" due to the forget gate in the LSTM architecture, meaning the word "ada" will be disregarded, and only the phrase "berapa kaki di seekor bebek" will be processed. However,

the BART and BERT methods scored zero due to the same lack of training data as before. For the fifth test question "bebek memiliki berapa kaki," all methods scored zero, including the proposed DNN method, due to the difficulty in predicting the correct answer because the test question contains words relevant not only to one or two corpus questions. For instance, the corpus questions include "berapa banyak kaki di seekor bebek," "buah apa yang memiliki duri," and "pelikan bisa hidup berapa lama." Thus, the test question "bebek memiliki berapa kaki" might be interpreted by the DNN method as "berapa banyak kaki di seekor bebek," where "memiliki" could be read as "buah apa yang memiliki duri," and the word "berapa" could be detected as part of the question "pelikan bisa hidup berapa lama." LSTM also failed to answer correctly due to the different structure of the test question, which differs from how LSTM processes words in a time series, while BART and BERT struggled because the test question's structure differed from that of the corpus questions.

For the second corpus question, "apa itu rusa," when tested with the first test question, which is exactly the same as the corpus question, all methods scored one because, for the DNN method, as long as the words provided are relevant to the corpus dataset, it is not an issue. The same applies to the LSTM method, where the test question is relevant to the corpus dataset and shares the same structure, so it is also not an issue. The same is true for the LSTM method as it is for the BART and BERT methods, which can understand the broader context and provide accurate answers.

In the second test, the question "rusa itu apa," the DNN method scored one as it could identify relevant keywords, despite the structure being different from the corpus training question structure. Meanwhile, LSTM struggled due to its limitations in processing structures different from the corpus dataset. Unlike LSTM and BART, which scored zero, BERT successfully provided the correct answer for the question "rusa itu apa" due to its ability to read the context in the test question. In the next question, "apa itu hewan rusa," DNN again scored one despite the addition of the word "hewan" in the question because the weights and biases during training were correctly assigned. However, the LSTM method scored zero, even though it has a mechanism to ignore irrelevant words in the corpus question, but in the

test question “apa itu hewan rusa,” even though the test question is almost the same as the corpus question, with only the added word “hewan,” LSTM could not answer correctly due to incorrect weight and bias assignment during training. The same applies as in the previous testing for the BART and BERT methods.

In the next test question “hewan rusa itu apa,” DNN answered correctly, even though the question included an additional word, “hewan,” at the beginning of the test question “hewan rusa itu apa,” compared to the corpus question “apa itu rusa,” because DNN recognized more relevant words and thus could predict the answer correctly. The LSTM, BART, and BERT methods struggled again in this case because the test question was not compatible with their respective methods, making it difficult to provide the correct answer. In the final testing with the question “apa hewan rusa itu,” the DNN method again scored one because the case is identical to the question “hewan rusa itu apa.” Likewise, the LSTM, BART, and BERT methods still scored zero because the test question provided does not match the pattern of the corpus question.

Based on the results obtained in Table VII, the DNN method demonstrates better capability in handling unstructured early childhood questions. However, this method may struggle to accurately predict the answer when the given question contains words relevant to two different corpus questions, as seen in the fifth test question, “bebek memiliki berapa kaki.” Despite this, DNN successfully predicted the correct answer for the fourth question, even with the additional word “ada” in the question “ada berapa kaki di seekor bebek.” In contrast, the LSTM method cannot provide the correct answer when the test question contains unstructured words or has a pattern that differs from the existing corpus questions. However, LSTM can manage questions with irrelevant additional words, thanks to its forget gate mechanism. The BART and BERT methods, on the other hand, have difficulties dealing with unstructured questions. In some cases, even when the test question is identical to the corpus question, these methods still face challenges in delivering the correct answer. Overall, the children asked a total of 2000 questions. The accuracy evaluation for the questions asked by early childhood children is as follows:

$$\text{DNN Accuracy} = \frac{1862+0}{1862+0+0+138} = \frac{1862}{2000} = 0.931 \times 100 = 93.1\%$$

$$\text{LSTM Accuracy} = \frac{1581+0}{1581+0+0+419} = \frac{1581}{2000} = 0.7905 \times 100 = 79.05\%$$

$$\text{BART Accuracy} = \frac{1286+0}{1286+0+0+138} = \frac{1286}{2000} = 0.643 \times 100 = 64.3\%$$

$$\text{BERT Accuracy} = \frac{1351+0}{1351+0+0+419} = \frac{1351}{2000} = 0.6755 \times 100 = 67.55\%$$

After conducting the answer response test, applying

DNN to 2000 questions yielded an impressive accuracy rate of 93.1%. Using LSTM resulted in a lower accuracy of 79.05%. Conversely, the BART and BERT methods struggled to process questions from early childhood children, resulting in the lowest accuracy rates for handling such questions, with BART achieving 64.3% and BERT 67.55%. Interestingly, the DNN method provided nearly 100% correct answers when the questions matched the question corpus. However, for the comparison methods, testing questions with the same pattern as the question corpus could still produce incorrect answers. These findings underscore the robustness and flexibility of the DNN model, particularly in handling novel or unconventional questions, highlighting its superiority over LSTM, BART, and BERT in the context of questions from early childhood children.

5. PRACTICAL IMPLICATIONS

The question-and-answer system addresses the challenge of recognizing questions from early childhood children, who often ask unstructured questions or use patterns that differ from the corpus questions, making it difficult for the system to evaluate accurately. For instance, the corpus question “apa itu rusa” is tested with various unstructured questions from early childhood children, such as “apa itu rusa,” “rusa itu apa,” “apa itu hewan rusa,” “hewan rusa itu apa,” and “apa hewan rusa itu.” The proposed DNN method is effective in recognizing these types of questions because it predicts answers based on relevant keywords. This allows it to provide accurate responses even when the test question structure differs from the corpus question. This question-and-answer system is also beneficial in helping children interact, as they can ask questions verbally, and the system will respond in the same manner. However, one of the challenges lies in cases where children ask questions that are not relevant to the corpus, leading the system to respond with “sorry, your question is not understood.” Another challenge arises when children struggle to form their questions, requiring guidance through stories or pictures to stimulate their curiosity.

For practical implementation, the question-and-answer system is introduced at the beginning of the class. If a child asks a spontaneous question, the teacher can encourage them to ask the system, and they will listen to the response together, making the learning process more interactive and fostering a closer relationship between the children and the system. This also enables children to continue asking questions to the system when the teacher is not present in the classroom. Additionally, questionnaires are used to gather feedback from both children and teachers on the system’s effectiveness, ensuring its continued improvement.

6. CONCLUSIONS

Based on the obtained results, it can be concluded that the findings from the training phase show that the DNN method consistently outperforms other methods in various training scenarios. For instance, between epochs 200 and 500, DNN achieved better loss values compared



to LSTM, BERT, and BART, with significant accuracy improvements. At epoch 500, DNN reached an accuracy of 93.59%, surpassing BERT (90.51%), BART (88.33%), and LSTM (88.24%).

The answer response test results demonstrate that the DNN method, when applied to 2000 questions, achieved an impressive accuracy rate of 93.1%. In comparison, the LSTM method yielded a lower accuracy of 79.05%. The BART and BERT methods struggled to process questions from young children, resulting in the lowest accuracy rates, with BART at 64.3% and BERT at 67.55%. The DNN method consistently provided correct answers when questions aligned with those in the corpus. However, unlike the comparison methods, test questions with similar patterns to corpus questions could still produce incorrect answers. These findings underscore the power and flexibility of the DNN model, especially in handling novel or unconventional questions, showcasing its superiority over LSTM, BART, and BERT for young children's questions. Nevertheless, the DNN method occasionally encountered challenges when irrelevant words appeared in the given question, such as "memiliki" in the testing question "bebek memiliki berapa kaki," compared to the dataset question "barapa banyak kaki di seekor bebek." Future research should aim to address this limitation.

REFERENCES

- [1] S. Perazzoli, J. P. de Santana Neto, and M. J. M. B. de Menezes, "Systematic analysis of constellation-based techniques by using natural language processing," *Technological Forecasting and Social Change*, vol. 179, 2022.
- [2] P. M. Mah, I. Skalna, and J. Muzam, "Natural language processing and artificial intelligence for enterprise management in the era of industry 4.0," *Applied Sciences*, vol. 8, 2022.
- [3] M. Shymbayev and Y. Alimzhanov, "Extractive question answering for kazakh language," *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, pp. 401–405, 2023.
- [4] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, and Y. Zhang, "Conversational question answering: a survey," *Knowledge and Information Systems*, vol. 64, pp. 3151–3195, 2022.
- [5] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multi-media tools and applications*, 2023.
- [6] S. Skandan, S. Kanungo, S. Devaraj, S. Gupta, and S. Narayan, "Question answering system using knowledge graphs," *2023 International Conference on Inventive Computation Technologies (ICICT)*, pp. 656–661, 2023.
- [7] A. Dwi Utomo, Z. Zainuddin, and S. Syarif, "Question answering system in the domain of early childhood education in bahasa indonesia," *2020 6th International Conference on Science and Technology (ICST)*, vol. 1, pp. 01–05, 2020.
- [8] B. Das and S. J. Nirmala, "Improving healthcare question answering system by identifying suitable answers," *022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, pp. 1–6, 2022.
- [9] D. Koshti, A. Gupta, and M. Kalla, "Knowledge blended open domain visual question answering using transformer," pp. 823–828, 2023.
- [10] G. Melissano, G. Tinelli, and T. Soderlund, "Current artificial intelligence based chatbots may produce inaccurate and potentially harmful information for patients with aortic disease," *European Journal of Vascular and Endovascular Surgery*, vol. 67, no. 4, pp. 683–684, 2024.
- [11] T. S. Sathe, J. Roshal, A. Naaseh, J. C. L'Huilier, S. M. Navarro, and C. Silvestri, "How i gpt it: Development of custom artificial intelligence (ai) chatbots for surgical education," *Journal of Surgical Education*, vol. 81, no. 6, pp. 772–775, 2024.
- [12] L.-Q. Cai, M. Wei, S.-T. Zhou, and X. Yan, "Intelligent question answering in restricted domains using deep learning and question pair matching," *IEEE Access*, vol. 8, pp. 32922–32934, 2020.
- [13] W. Budiharto, V. Andreas, and A. A. S. Gunawan, "Deep learning-based question answering system for intelligent humanoid robot," *Journal of Big Data*, 2020.
- [14] M. Therasa and G. Mathivanan, "Arnn-qa: Adaptive recurrent neural network with feature optimization for incremental learning-based question answering system," *Applied Soft Computing*, vol. 124, p. 109029, 2022.
- [15] H. Alami, A. El Mahdaouy, A. Benlahbib, N. En-Nahnahi, I. Berrada, and S. E. A. Ouatik, "Daqas: Deep arabic question answering system based on duplicate question detection and machine reading comprehension," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101709, 2023.
- [16] A. Adio and A. Simeon, "A linguistic variable of product-related question answering review system," *Systems and Soft Computing*, vol. 5, p. 200047, 2023.
- [17] J. Bissantz, J. Karpowski, M. Steinhausen, Y. Luo, F. Ferraro, A. Scholtissek, C. Hasse, and L. Vervisch, "Application of dense neural networks for manifold-based modeling of flame-wall interactions," *Applications in Energy and Combustion Science*, vol. 13, p. 100113, 2023.
- [18] N. Annamalai, R. A. Rashid, U. Munir Hashmi, M. Mohamed, M. Harb Alqaryouti, and A. Eddin Sadeq, "Using chatbots for english language learning in higher education," *Computers and Education: Artificial Intelligence*, vol. 5, p. 100153, 2023.
- [19] S. H. Alqaidi, S. M. Albugami, W. S. Alzahrani, S. Badri, and A. Wali, "Network-integrated medical chatbot for enhanced healthcare services," *Telematics and Informatics Reports*, vol. 15, p. 100153, 2024.
- [20] B. Alsafari, E. Atwell, A. Walker, and M. Callaghan, "Towards effective teaching assistants: From intent-based chatbots to llm-powered teaching assistants," *Natural Language Processing Journal*, vol. 8, p. 100101, 2024.
- [21] R. Olszewski, K. Watros, M. Mańczak, J. Owoc, K. Jeziorski, and J. Brzeziński, "Assessing the response quality and readability of chatbots in cardiovascular health, oncology, and psoriasis: A comparative study," *International Journal of Medical Informatics*, vol. 190, p. 105562, 2024.
- [22] J. Du and B. K. Daniel, "Transforming language education: A systematic review of ai-powered chatbots for english as a foreign

- language speaking practice,” *Computers and Education: Artificial Intelligence*, vol. 6, p. 100230, 2024.
- [23] K. Guo and D. Li, “Understanding efl students’ use of self-made ai chatbots as personalized writing assistance tools: A mixed methods study,” *System*, vol. 124, p. 103362, 2024.
- [24] D. A. Siddiqi, F. Miraj, H. Raza, O. A. Hussain, M. Munir, V. K. Dharma, M. T. Shah, A. Habib, and S. Chandir, “Development and feasibility testing of an artificially intelligent chatbot to answer immunization-related queries of caregivers in pakistan: A mixed-methods study,” *International Journal of Medical Informatics*, vol. 181, p. 105288, 2024.
- [25] M. A. O. Zishan, H. Shihab, S. S. Islam, M. A. Riya, G. M. Rahman, and J. Noor, “Dense neural network based arrhythmia classification on low-cost and low-compute micro-controller,” *Expert Systems with Applications*, vol. 239, p. 122560, 2024.
- [26] A. Babu and S. B. Boddu, “Bert-based medical chatbot: Enhancing healthcare communication through natural language understanding,” *Exploratory Research in Clinical and Social Pharmacy*, vol. 13, p. 100419, 2024.
- [27] U. Hahn and T. Duan, “Corpus assembly as text data integration from digital libraries and the web,” *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 25–28, 2019.
- [28] L. Zahrotun, N. H. Putri, and A. Nur Khusna, “The implementation of k-means clustering method in classifying undergraduate thesis titles,” pp. 1–4, 2018.
- [29] Z. Sultana Ritu, N. Nowshin, M. Mahadi Hasan Nahid, and S. Ismail, “Performance analysis of different word embedding models on bangla language,” vol. 17, pp. 1–5, 2018.
- [30] E. Ferreira, B. Jabaian, and F. Lefèvre, “Online adaptative zero-shot learning spoken language understanding using word-embedding,” *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5321–5325, 2015.
- [31] F. Nazari and W. Yan, “Convolutional versus dense neural networks: Comparing the two neural networks performance in predicting building operational energy use based on the building shape,” *ArXiv*, vol. abs/2108.12929, pp. 1–6, 2021.
- [32] S. Agatonovic-Kustrin and R. Beresford, “Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research,” *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717–727, 2000.
- [33] S. K. Bashar, A. A. Fahim, and K. H. Chon, “Smartphone based human activity recognition with feature selection and dense neural network,” pp. 5888–5891, 2020.
- [34] Y. Tian, Y. Zhang, and H. Zhang, “Recent advances in stochastic gradient descent in deep learning,” *Mathematics*, vol. 11, no. 3, 2023.
- [35] S. G. A, R. Prabha, A. M, A. S. A. Nisha, and S. Prabu, “A novel predictive analysis and classification of land subsidence vulnerability mapping based on gis using hybrid optimized machine learning techniques and computer vision,” *Procedia Computer Science*, vol. 233, pp. 343–352, 2024.
- [36] K. Peyton and S. Unnikrishnan, “A comparison of chatbot platforms with the state-of-the-art sentence bert for answering online student faqs,” *Results in Engineering*, vol. 17, p. 100856, 2023.
- [37] L. Zhao, Z. Shang, L. Zhao, A. Qin, and Y. Y. Tang, “Siamese dense neural network for software defect prediction with small data,” *IEEE Access*, vol. 7, pp. 7663–767, 2019.
- [38] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, pp. 25–28, 2014.
- [39] D. B. Tariang, R. S. Chakraborty, and R. Naskar, “A robust residual dense neural network for countering antiforensic attack on median filtered images,” *IEEE Signal Processing Letters*, vol. 26, pp. 1132–1136, 2019.