International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

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

La Ode Fefli Yarlin[1], Zahir Zainuddin[1*], and Ingrid Nurtanio[1]

[1] Departement of Informatics, Hasanuddin University, Makassar, Indonesia

E-mail address: yarlinlof21d@student.unhas.ac.id, zahir@unhas.ac.id, ingrid@unhas.ac.id

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Questions are a well-known topic in Natural Language Processing (NLP). This feature is very suitable for use in learning activities in kindergarten to help train social interaction. The problem in this research is that the developed system must be able to understand questions from childhood. This is complex, given that their questions often need to be spoken correctly due to their limited ability to formulate questions appropriately. Therefore, this research proposes the Dense Neural Network (DNN) method, which can handle questions with non-linear word order using an Indonesian corpus of 5000 questions and answers. Experimental results show that the proposed DNN approach is superior to the Long Short Term Memory (LSTM) method in understanding and answering questions from young children, especially those that need to be more structured and formulated but have a clear context. DNN also achieved the highest accuracy in the training process, which was 0.9356. In contrast, the LSTM method showed a lower accuracy of only 0.8824. In a test of 2000 questions with different question patterns, the best accuracy was obtained by the DNN method at 93.1\%. The results of this study make an essential contribution to the development of NLP systems that can be used in the context of early childhood learning.

Keywords: Dense Neural Network (DNN), Long Short Term Memory (LSTM), Natural Language Processing (NLP), Question and answer

1. Introduction

Natural Language Processing (NLP) is essential in the advancement of Artificial Intelligence (AI), which aims to create systems that can process human language at a similar or higher level than humans, including tasks such as understanding speech text, analyzing, categorizing, translating, and more [1]. Question answering (QA) is an essential task in the field of NLP and Information Retrieval (IR) in our research, which aims to provide answers to natural language questions using unstructured documents [2][3]. The problems that often occur in natural language processing are illogical words, such as nouns and verbs given as axioms. The number of illogical words is vast, and their semantics depend on the problem domain, which is difficult to define because the system needs to understand them [4]. Sometimes, data limitations can also affect the model's ability to understand and produce text well.

In early childhood, speech develops rapidly. Therefore, one of the characteristics of early childhood is the penchant for asking questions. Children often ask questions about everything they see and think, and from there, conversations are formed. However, obstacles to developing children's speaking skills must be achieved optimally. Along with technological advances, many innovations have been developed to help children's growth and development, including technology in education to facilitate more exciting learning.

One of the learning tools currently developing is an educational system tool. As the feature is a question-answering system, question-and-answer is a well-known topic in NLP [5]. This feature is very suitable for use in learning activities in kindergarten to help train social interaction. A question-answering system is a model that answers questions posed in natural language by extracting relevant answers from a paragraph [6]. Question-answering systems are also a technology needed today to facilitate automating user answers [7]. This is shown by a large amount of research on Automated QA question and answer systems, considered one of the leading research topics because the machine must understand the context

E-mail: yarlinlof21d@student.unhas.ac.id, zahir@unhas.ac.id, ingrid@unhas.ac.id

of questions in various fields and languages, including Indonesian [8]. Getting the most appropriate and precise answer to a given question [5] is still the biggest challenge in question-and-answer systems because it requires incorporating human intelligence into the machine [9].

Educational devices with a question-and-answer system, where questions are input through speech and answers are output through speech, can help children increase their knowledge and practice communication skills. This helps achieve the goals of early childhood education. Research on assistive learning media for early childhood is still rare, especially question-and-answer systems integrated into educational tools. This integration can affect children's development. One important aspect is the question domain to build a question-and-answer system in early childhood. Young children cannot direct questions correctly because their articulation needs improvement.

The challenge in this research lies in developing a system that can understand questions from young children. This is complex, considering their questions often need to be pronounced correctly due to articulation constraints and their lack of ability to formulate questions well. In addition, the system also needs to be able to interpret the context of the questions the child asks. The question-and-answer system being developed aims to provide accurate answers to these questions. Several methods have been used in Natural Language processing, especially in question-and-answer systems. However, challenges include inputting words that do not match the problem domain and not understanding the context of the question [10]}.

2. RELATED WORK

Related research that discusses the question-answering system using computer vision entitled Automated Thai-FAQ Chatbot using RNN-LSTM utilizes the Recurrent Neural Network (RNN) method and the Long-Short Term Memory (LSTM) method for answer selection on automatic chatbots. The research got an accuracy from system testing of 93.2\% for correct answers from 86.36\% of processed questions, and 13.64\% of questions were ignored because the correct answer was not found [11].

Furthermore, research that uses a Recurrent Neural Network (RNN) equipped with a Long-Short Term Memory (LSTM) gate mechanism to develop a question-and-answer system entitled Question Answering System in the Early Childhood Education Domain in Indonesian. The research conducted by [7] reported the accuracy values obtained from system testing. Specifically, the RNN achieved an accuracy rate of 78.11\%, while the LSTM showed superior performance with 89.5\% accuracy.

Research [10] also discusses a question and answer system entitled Intelligent Question Answering in Restricted Domains Using Deep Learning and Question Pair Matching Using CNN and BiLSTM methods with problems Chinese sentences are more complex and diverse in expression, and there are also specific difficulties in parsing word meanings and coding restricted domains. The study got an accuracy of 86.38\% from system testing. CNN is used to capture important feature information, while the BiLSTM network is used to obtain a semantic analysis of the entire sentence.

Furthermore, a related study concerning the research is titled Deep Learning-based Question Answering System for an Intelligent Humanoid Robot. Delves into utilizing extensive datasets to furnish the model with information, constructing a knowledge repository from vast textual datasets, and amalgamating a database with Artificial Intelligence Markup Language (AIML) to scrutinize missing data formats or information required to formulate responses in the chatbot. The article also explores the application of deep learning methodologies, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), tailored explicitly for sequential data processing [12].

The subsequent research also discusses a question-and-answer system titled A Chinese Oral Question-and-Answering System Based on LSTM. This study presents a question-and-answering system based on deep learning neural network methods, specifically KALDI and MATCH-LSTM. The system has been implemented on desktop and embedded platforms, with promising results indicating the approach's effectiveness. The system can recognize the main keywords in questions and provide accurate answers, even when voice recognition has an error rate of 17%. The system's accuracy rate, as evaluated by the BLEU-4 method, is 31.4%, which is 23% lower than human answers [13].

Furthermore, the research titled A Spoken-Based Question Answering System for Train Route Service using the Frame-Based Approach discusses the question-answering system. This study paper presents the development of a speech-based question-answering system for train route service using a frame-based approach. The system underwent a thorough evaluation concerning speech recognition and linguistic comprehension, whereby the automatic speech recognition (ASR) component attained an average word accuracy of 80.53%. In comparison, the natural language understanding (NLU) component exhibited a perfect accuracy rate of 100%. The ASR module's precision was attained through many techniques, encompassing hidden Markov models, keyword spotting, and pre-scripted responses [14].

Finally, this research is a reference for the research to be carried out, but this research is different from the research to be carried out. However, this study uses the method used in the current study because it is superior to previous studies' accuracy. This research uses the Dense Neural Network (DNN) method to detect cardiac arrhythmia and implements it on a low-cost, power-efficient, low-computation ECG monitoring system. The results show that the developed system successfully detects arrhythmia with an accuracy of about 97.09\%. A cardiac arrhythmia detection algorithm through an artificial neural network was used in this research. The system successfully detects arrhythmia with a small model size, making it suitable for use on MCUs while maintaining good performance [15].

Research conducted by Utomo studied the utilization of the Long Short-Term Memory (LSTM) method [5], distinguished for its capacity to meticulously parse questions linearly, meticulously adhering to the sequential arrangement of words without any deviations or omissions. However, this precise modus operandi imposes a noteworthy constraint: LSTM operates optimally when the queries posed align harmoniously with the pre-established structures within the dataset. Consequently, this rigidity in approach renders LSTM less adept at handling inquiries that exhibit parallel contextual themes yet adopt varying syntactic constructions. This inherent inflexibility directly impacts the fidelity and precision of the responses generated, underscoring a notable limitation in LSTM adaptability across diverse linguistic frameworks and question formats.

In contrast, the notable contribution of this research lies in the application of Deep Learning algorithms, particularly the Dense Neural Network (DNN) method. Unlike LSTM, these algorithms do not directly process individual words but necessitate the conversion of words within the corpus dataset into vectors before proceeding with analysis. Deep Learning algorithms leverage these vector representations to conduct their operations. What distinguishes the DNN method is its remarkable capability to comprehend inquiries posed by young children, even when they exhibit irregular or unstructured formats. Unlike LSTM, which is bound by a linear word order constraint, DNN exhibits a unique proficiency in accommodating questions with non-linear word arrangements, thus granting the system enhanced flexibility and precision in grasping and responding to queries from young children. This inherent adaptability positions DNNs as more suitable and effective than LSTM in scenarios where linguistic structures may deviate from conventional norms, particularly in contexts involving youthful language acquisition and comprehension.

3. METHODOLOGY

This research began by creating an early childhood education-specific dataset containing approximately 5000 Indonesian-language questions. This dataset was developed by collecting conversations among early childhood students from various sources, including early childhood schools, educational books, and online sources. The dataset is unique in that it contains questions and answers, creating question-answer pairs that will be used to train the model. The model development process went through several key stages. The first stage is the collection and organization of this dataset or corpus. Once the dataset is ready, the research moves to the second stage, word tokenization. This stage involves breaking the text into smaller pieces, or "tokens", so the computational model can process them. Next, in the third stage, a word embedding process is performed, where these tokens are converted into numerical vectors that can represent their semantic meaning in a multidimensional space.

The next stage, the fifth stage, involves using the results of the embedding process in a Dense Neural Network (DNN). In this stage, the DNN model is trained using a prepared corpus of questions and answers to develop a model that can accurately understand and respond to questions. The DNN training process involves an iterative process where the model gradually adjusts its internal parameters to improve its predictive ability and understanding of human language. The result is a model that can respond appropriately and contextually to different types of questions asked. The details are shown in Fig. 1 and included in the study to understand this process better.

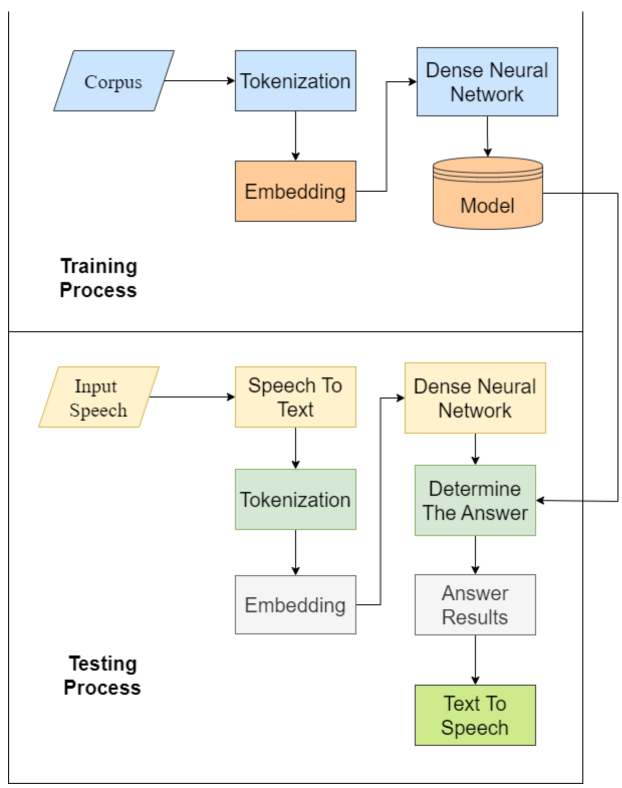

Figure 1. Blok diagram of the Question and Answer System

In this research, the developed question-and-answer system will be tested through various scenarios, each with a unique question pattern. These scenarios include structured questions, semi-structured questions, and unstructured questions. This approach aims to evaluate the system's ability to handle different types of questions, varying from those with a fixed and clear format to those that are more free-form and unorganized. By performing these scenarios, it can be seen how effective the system is in processing and providing accurate answers under various conditions. More details regarding the scenarios are shown in Fig. 2.

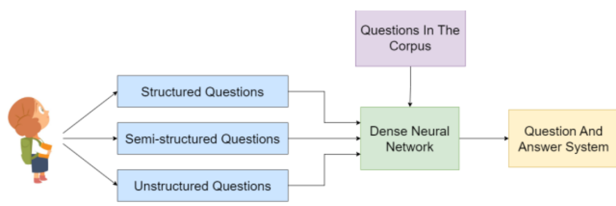

Figure 2. Block Diagram of the Scenario to be Performed

In this research scenario, three types of tests will be used, where young children will ask several questions. The questions will be divided into three types: structured, semi-structured, and unstructured, but still have a similar context. Next, the questions will be classified using the Dense Neural Network method to find a match with the questions in the corpus. The answer will be output if a match is found with the question in the corpus. Moreover, by conducting this set of scenarios, this research can provide a comprehensive picture of how well the system can adapt and perform in different situations that may differ. Through testing different question types and linguistic variations, this research aims to evaluate the extent to which the system can overcome its challenges. This thorough evaluation is essential to identify the strengths and weaknesses of the system to guide further development and improvement.

A. *Preprocessing Stage*

1) *Corpus*

A corpus collects important text or language data for linguistic analysis or language technology development. New approaches to corpus construction maximize the coverage of targeted resources by collecting fragments from various sources [16]. In this study, the constructed corpus specifically focuses on the Indonesian language, particularly regarding questions and answers originating from early childhood. The corpus includes 5000 question-and-answer data pairs collected and organized to create a rich and diverse database. After the corpus-building process is complete, the next step is the tokenization stage. In the tokenization stage, the text in the corpus will be broken down into smaller units called tokens, such as words, symbols, or other language elements. This process is important because it allows the system to easily understand and process the natural language used in questions and answers. This tokenization is an important step in processing natural language data for further preparation in the Deep Learning process. More details about the composition and structure of the corpus used are shown in Tab. 1.

TABLE I. Sample corpus of questions and answers in Indonesian

Question	Answering
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

Tab. 1 shows an example of the corpus data that we use in the table, and there are 2 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 "Hey friend," "Good morning," "Good night," "What is your name?" and the last, "Why don't you have a name?" in English, the questions are "Hi friend," "Good morning," "Good night," "What is your name?" The answer column consists of five answers in Indonesian, namely "Hi there, friend," "Good morning too," "Good night too," "I do not have a name," and the last, "Because I am a 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 [17]. 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 Fig. 3.

```
'aku', 'kamu', 'nama', 'punya', 'siapa', 'tidak'
```

Figure 2. Example of Indonesian Tokenization Results

The tokenization result shown in Fig. 3 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 [18]. 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 [19]. 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. Fig. 4 shows the results of embedding the question-and-answer words into vectors.

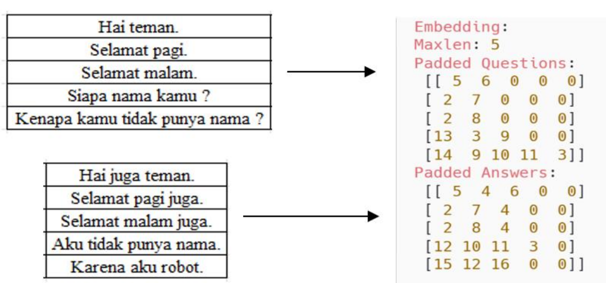

Figure 2. Embedding Results of Questions and Answers

Fig. 4 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)

Dense Neural Network (DNN) is the simplest neural network architecture. It is a biologically inspired computational model designed to mimic how the human brain processes information [20][21]. DNN consist of multiple hidden layers, each consisting of several fully connected or "dense" neurons. Each layer contains a predetermined number of neurons removed from one layer that serve as inputs to the next layer [22]. DNN can analyze complex data patterns and have been used in various disciplines for diverse applications, ranging from regression analysis and classification to unsupervised data clustering. The compact mathematical equation form of a Dense Neural Network is as follows.

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

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.

The ReLU activation function was used in this study's dense neural network method. This function was chosen due to its ability to overcome the missing gradient problem and faster convergence speed in neural network training. The following is the ReLU function formula used in the first and second hidden layers in dense neural networks:

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

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 less than or equal to zero will be converted to zero, while values greater than zero will be retained. This allows 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 DNN 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 [15]. By using Softmax, the model can provide a prediction of the most likely class for each instance of input data, and it is important to understand how a DNN translates its output into a classification task.

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

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.

The object of assessment is the value of the accuracy matrix. The formula for calculating the accuracy of the system is 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.

The accuracy matrix is used to evaluate how well the system can classify the data correctly, measuring the proportion of correct predictions out of the overall data tested. The Dense Neural Network (DNN) architecture is shown in Fig. 5.

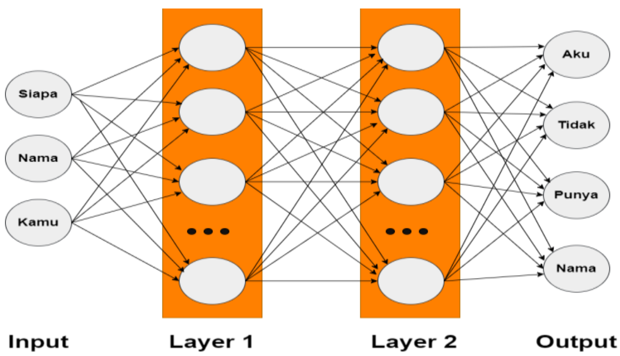

Figure 5. Dense Neural Network (DNN) Architecture

At this stage, the DNN method will receive the preprocessing results for the training process, and the DNN method here will process the vector results of the question. For example, we will process the "siapa nama kamu" question with vector values “siapa” = 13, “nama” = 3, and “kamu” = 9. The question has three vector values, so there will be three inputs in the input part of the architecture.w For example, for the first stage, vector 13 will be inserted into the first hidden layer neuron with 256 neurons to calculate the weight and bias. Each neuron in the layer has a weight and a bias. The mathematical equation is z = $\sigma\ (W \cdot x + b\ )$.

Input Layer x_1, z = output value of the first hidden layer with activation function = σ with $(W_1, W_2, W_3 \ldots . W_n)$ from each existing weight it will be multiplied by the input x_1 after obtaining the result of each multiplication after that it will be added to each existing weight, namely $(b_1, b_2, b_3 \ldots b_n)$ the result will use the Relu activation function to get a definite value and will be output z. the output of the first layer will be the input of the second hidden layer, at this stage a stage will be carried out as in the first hidden layer. The output of the second hidden layer will be rounded with Relu. Back then, it would be used again. The Softmax activation function would solve the code by classifying the answer by studying the output relationship between the results of the three inputs by getting a value of 1 if the answer matches the corpus data with the testing data. If not, then the answer will get a value of 0. The output of this layer is a response to the question given in the form of an answer.

2) Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) introduced in 1997 by Hochreiter and Schmidhuber. LSTMs are designed to learn long-term dependencies by using gates to control the flow of information into and out of cell states. LSTM consists of four interacting layers in each repeating module: forget gate, input gate, cell vector, new cell state, output gate, and LSTM output. LSTM overcomes problems such as long-term dependency, gradient loss, and gradient explosion commonly found in traditional RNNs. With the ability to store information over long periods, LSTMs outperform traditional RNNs [23]. The architecture of Long Short-Term Memory (LSTM) is shown in Fig. 6.

In this LSTM method, the architecture comprises two distinct components: the LSTM encoder functions as the input, while the LSTM decoder serves as the output. Prior to initiating the training process, LSTM receives preprocessing data from the preceding stage, namely the embedding phase, wherein words are converted into vector representations. This preprocessing step is essential as LSTM inherently cannot directly process words. To illustrate, let us consider the processing of the same question, "siapa nama kamu," using vector values such as "siapa" = 13, "nama" = 3, and "kamu" = 9. With three vector values corresponding to the three words in the question, the LSTM model incorporates three inputs along with three LSTM gates to facilitate the processing of the input sequence.

For example, vectors 13, 3, and 9 will be inserted into the hidden layer neurons of the three LSTM gates with 256 neurons for the first stage to calculate weights and biases. If the results obtained in the layer are relevant to the existing data, the data will be forwarded to the second gate, namely the LSTM decoder gate; otherwise, if the vector value 13 is not relevant to the existing data, the data will not be forwarded or forgotten. The relevant data will be processed through the Relu activation function to help learn the data relationship in sequence. The output of the hidden layer of the LSTM will be the input for the second hidden layer, where, at this stage, a similar process will be carried out in the first layer. The output of the second hidden layer is forwarded to the output layer. This part is the final process, which is precisely the same as the previous DNN method using the Softmax activation function for classification tasks, unlike the previous activation function, Relu; the output of this layer is the response to the given question in the form of an answer.

International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

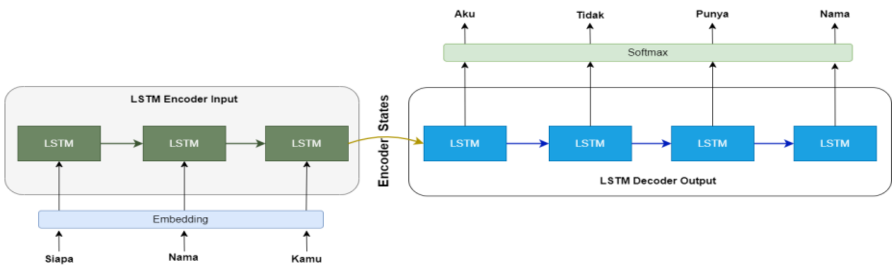

Figure 6. Long Short Term memory (LSTM) architecture

4. RESULT AND DISCUSSION

In the results and discussion section, we will initially convert the questions and answers from the Text format to the JSON format for the corpus dataset. This conversion is undertaken because previous research [7] utilized the text format to structure questions and answers. However, to achieve a variety of questions and answers, it is necessary to create multiple instances of each. Therefore, this study will employ a different format, namely the JSON format. This choice is informed by research [24] that demonstrates the JSON format's capability to accommodate multiple patterns of questions and answers within a single question tag, thereby enhancing efficiency and diversity in the representation of questions and answers. For example, we will take a question in Indonesian, "Selamat pagi," which translates to "Good morning" in English. The answer can have several patterns, such as "Selamat pagi," "Pagi juga," and "Pagi". In English, these correspond to "Good morning," "Good morning to you," and simply "Morning". Fig. 7 shows the result of converting the text corpus data set to JSON.

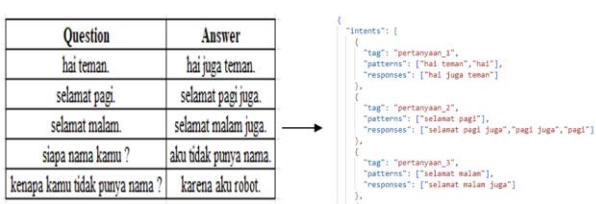

Figure 7. Text Corpus Dataset Conversion to JSON

In the training process, the 5000 question-and-answer corpus datasets will be meticulously configured into hyperparameters for both the DNN and LSTM algorithms. These hyperparameters will encompass a range of epoch values spanning from 100 to 500 epochs while simultaneously integrating 2 layers into the architecture. Additionally, the training process will entail configuring the batch size to a fixed value of 16, alongside adjusting other pertinent settings to optimize model performance. More detailed information on these settings can be found in Tab. 2 for a deeper understanding.

TABLE II. Hyper Parameter Settings

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

In Tab. 2, a comprehensive depiction of the training process scenario is provided. The input layer optimizer selected for this research is Adaptive Moment Estimation (Adam), a fitting choice considering the extensive dataset employed. The learning rate for the Adam optimizer is set at 0.01, with a consistent batch size of 16 across the varying epochs, ranging from 100 to 500. In the LSTM method, each layer is equipped with 256 nodes, incorporating the Rectified Linear Unit (ReLU) activation function. To mitigate issues of overfitting and underfitting, a dropout rate of 0.5, considered an optimal middle ground, is applied [25][26]. Furthermore, the softmax activation function is utilized to classify the output results issued by the LSTM model.

In the DNN scenario, a parallel approach to the LSTM method scenario will be adopted. The training will encompass a series of processes, each employing varying epoch values alongside a fixed batch size of 16. Specifically, the first process will utilize epoch 100, followed by the second process with epoch 200, the third process with epoch 300, the fourth process with epoch 400, and finally, the fifth process with epoch 500. These sequential processes are executed to meticulously evaluate the accuracy of the hyperparameter configurations when

E-mail: yarlinlof21d@student.unhas.ac.id, zahir@unhas.ac.id, ingrid@unhas.ac.id

applied to the 5000 corpus datasets comprising early childhood questions and answers. The outcomes of this comprehensive scenario are encapsulated in Tab. 3, providing a detailed insight into the performance metrics and effectiveness of the DNN model under differing training configurations.

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

Methode	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

In Tab. III, the experimental findings underscore the superior performance of the proposed DNN approach compared to the LSTM method [27]. Across various scenarios, DNN consistently outperforms LSTM. For instance, in the initial scenario with an epoch value of 100 and a batch size of 16, LSTM yields an accuracy of 0.4792 with a loss of 1.4006. At the same time, DNN exhibits a slight improvement, achieving an accuracy of 0.5048 with a loss value of 1.9855.

This trend persists throughout the experimentation, culminating in the final scenario with epoch 500 and batch size 16, where DNN maintains its superiority over LSTM. Here, DNN achieves an impressive accuracy of 0.9356 with a loss value of 0.1809, while LSTM lags with an accuracy of 0.8824 and a loss of 0.3820. These consistent results are validated through multiple trials, affirming the robustness of the DNN model. For a comprehensive understanding of the accuracy value comparison, Fig. 8 presents a graphical representation, offering further insights into the performance disparities between DNN and LSTM across varying training scenarios.

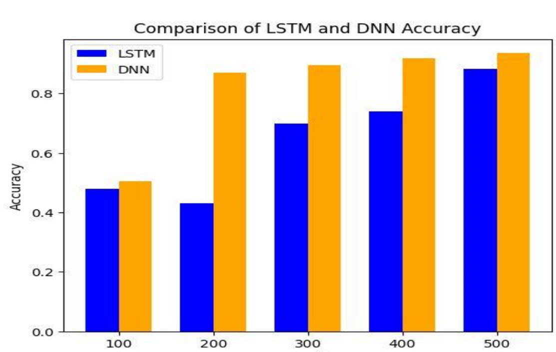

Figure 8. Comparison Graph of DNN and LSTM Accuracy

Following the training phase, the testing stage will be conducted in an early childhood environment. In this stage, children will interact with the system by posing questions verbally, which will then be transcribed into text format. Subsequently, text preprocessing will be executed in preparation for the subsequent stage, namely the answer classification stage. During this phase, the trained model obtained from both the training and testing processes will be deployed to discern appropriate answers. The system will then respond to the queries by articulating answers in spoken form once again. Table 4 exemplifies this process by presenting an instance of a question posed by an early childhood child to the system, alongside the corresponding answer response generated by the system for the child's inquiry. This example serves to illustrate the interaction between the system and the young user, demonstrating the efficacy of the model in understanding and responding to queries conversationally.

Tab. IV shows the test results of the question-and-answer system for early childhood, including the answers given. For example, in the first question, “Berapa kaki bebek”, the question pattern is different from the corpus dataset question, which is “Berapa banyak kaki pada seekor bebek”. In the DNN method, the system answered correctly, earning a score of 1, while the LSTM only earned a score of 0 because the answer was wrong. This error occurred because the question had a missing word, which did not match the dataset. The same thing happened in the second test. However, in the third and fourth tests, DNN and LSTM scored 1 because the questions given in the third test were from the dataset.

In contrast, LSTM could score one on the fourth test because the questions given were almost by the existing dataset. However, there were additional words “Ada” at the beginning of the question ”Berapa banyak kaki di seekor bebek” This happens because LSTM has a function to forget or forget what if there are words that are not relevant to the existing dataset. In the fifth test, for the question “Bebek memiliki berapa kaki”, DNN gave the wrong answer score 0 because, in the dataset, there is more than one question about ducks, such as “Bebek memiliki suara seperti apa”, so DNN are confused about choosing the correct answer. Meanwhile, LSTM gave the wrong answer because the questions were not in order and did not match the dataset. In contrast to the eighth test result, where LSTM obtained a score of 1 on the question “Apa itu rusa”, which has the additional word “Hewan” between the word “Itu” and the word “Rusa”. This happens because LSTM has a “Forget” function, which allows it to ignore words that are not relevant to the dataset. To calculate the accuracy of the answers, we use the accuracy formula where the total correct results are divided by the number of questions, then multiplied by 100.

TABLE IV. Example of Comparative Test Results between Methods Applied to a Question-and-Answer System Aimed at Young Children

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

$$DNN\ Accuracy = \frac{1862+0}{1862+0+0+138} = \frac{1862}{2000}$$
$$= 0.931 \text{ x } 100 = 93{,}1\%$$

$$LSTM\ Accuracy = \frac{1581+0}{1581+0+0+419} = \frac{1581}{2000}$$
$$= 0.7905 \text{x } 100 = 79{,}05\%$$

Subsequent to conducting the answer response test, employing DNN on a set of 2000 questions yielded an impressive accuracy rate of 93.1%. Conversely, utilizing LSTM resulted in a comparatively lower accuracy of 79.05%. Remarkably, questions that directly corresponded to the dataset achieved a perfect accuracy rate of 100% for both methods. However, when presented with newly generated questions, DNN exhibited superior adaptability, successfully recognizing and responding to the inquiries, whereas LSTM encountered difficulties and required assistance in identifying the provided questions. These findings underscore the robustness and flexibility of the DNN model, particularly in handling novel or unconventional queries, thus highlighting its superiority over LSTM in this context.

5. CONCLUSION

This study proposes the Dense Neural Network (DNN) method to develop an early childhood question-and-answer system. This method shows a remarkable ability to understand questions from children. Although the questions could be better formulated or more structured, they still have a clear context and can be understood. This is in contrast to the results obtained from previous research using the Long Short-Term Memory (LSTM) method, where LSTM is only able to recognize questions that have a sequential or time-series structure and relies heavily on the order of the words in the question to match the existing corpus dataset.

Based on the training outcomes, it is evident that the DNN method outperformed LSTM, demonstrating superior performance. DNN achieved the highest accuracy score of 0.9356 with a loss value of 0.1809, showcasing its efficacy in accurately classifying data. In contrast, the LSTM method exhibited a lower accuracy rate of 0.8824, coupled with a higher loss value of 0.3820, indicating its comparatively less effective performance in capturing underlying patterns within the dataset. In a subsequent test involving 2000 questions characterized by diverse question patterns, DNN exhibited remarkable proficiency by correctly answering 1862 questions, resulting in an impressive accuracy rate of 93.1\%. Conversely, in the same test, LSTM only managed to answer 1581 questions correctly, yielding a notably lower accuracy of 79.05\%. These results underscore the significant advantages of employing the DNN method, particularly in comprehending and addressing a wide range of queries, especially within the domain of early childhood language processing and understanding.

REFERENCES

[1] 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.

[2] 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.

[3] B. Ojokoh and E. Adebisi, “A review of question answering sys-tems,” Journal of Web Engineering, vol. 17, pp. 717–758, 2018.

[4] H. Seki, E. Nabika, T. Matsumura, Y. Sugiyama, M. Fujii, K. Torii, and T. Kasami, “A processing system for programming specifica-tions in a natural language,” vol. 2, pp. 754–763, 1988.

[5] P. Biswas, A. Sharan, and R. Kumar, “Question classification using syntactic and rule based approach,” pp. 1033–1038, 2014.

E-mail: yarlinlof21d@student.unhas.ac.id, zahir@unhas.ac.id, ingrid@unhas.ac.id

[6] S. Skandan, S. Kanungo, S. Devaraj, S. Gupta, and S. Narayan, "Question answering system using knowledge graphs," 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," vol. 1, pp. 01–05, 2020.

[8] B. Das and S. J. Nirmala, "Improving healthcare question answering system by identifying suitable answers," 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] 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. 32 922–329 341, 2020.

[11] P. Muangkammuen, N. Intiruk, and K. R. Saikaew, "Automated thai-faq chatbot using rnn-lstm," pp. 1–4, 2018.

[12] W. Budiharto, V. Andreas, and A. A. S. Gunawan, "Deep learning-based question answering system for intelligent humanoid robot," Journal of Big Data, 2020.

[13] K.-H. Chen, J.-I. Kao, C.-H. Wu, Y.-R. Chen, and Y.-C. Huang, "A chinese oral question-and-answering system based on lstm," pp. 1–2, 2019.

[14] D. P. Lestari and R. R. Nugraha, "A spoken-based question answer-ing system for train route service using the frame-based approach," pp. 1–6, 2017.

[15] 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. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S0957417423030622.

[16] U. Hahn and T. Duan, "Corpus assembly as text data integration from digital libraries and the web," pp. 25–28, 2019.

[17] 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.

[18] Z. Sultana Ritu, N. Nowshin, M. Mahadi Hasan Nahid, and S. Is-mail, "Performance analysis of different word embedding models on bangla language," vol. 17, pp. 1–5, 2018.

[19] E. Ferreira, B. Jabaian, and F. Lef`evre, "Online adaptative zero-shot learning spoken language understanding using word-embedding," pp. 5321–5325, 2015.

[20] 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.

[21] 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. [Online]. Available:https://www.sciencedirect.com/science/article/ pii/S0731708599002721

[22] 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.

[23] P. Choudhary and S. Chauhan, "An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism," Decision Analytics Journal, vol. 9, p. 100359, 2023. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S2772662223001996

[24] K. Peyton and S. Unnikrishnan, "A comparison of chatbot platforms with the state-of-the-art sentence bert for answering online student faqs," vol. 17, p. 100856, 2023.

[25] 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.

[26] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," vol. abs/1412.6980, pp. 25–28, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[27] D. B. Tariang, R. S. Chakraborty, and R. Naskar, "A robust residual dense neural network for countering antiforensic attack on median filtered images," vol. 26, pp. 1132–1136, 2019.

La Ode Fefli Yarlin and a short biography is a student at Hasanuddin University of Makassar, Indonesia. He received his bachelor's degree in informatics engineering from Dayanu Ikhsanuddin University, Baubau, Indonesia in 2020. His research interests focus on Natural Language Processing (NLP).

Zahir Zainuddin holds a Doctor of Computer Engineering from Bandung Institute of Technology, Indonesia, in 2004. He also received his B.Sc. in Electrical Engineering Department, Hasanuddin University, Indonesia, in 1988 and his M.Sc. (Computer Engineering) from Florida Institute of Technology USA in 1995. He is an associate professor at the Department of Informatics at Hasanuddin University in Indonesia. His research includes Computer Systems, intelligent systems, computer vision, and smart cities. He has published over 60 papers in international journals and conferences. In 1989, he was a JSPS research fellow at the Tokyo Institute of Technology.

Ingrid Nurtanio received the bachelor degree in Electrical Engineering from Hasanuddin University, Makassar, Indonesia in 1986. She received her Master of Technology from Hasanuddin University, Makassar, Indonesia in 2002. She received her doctoral degree from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia in 2013. Her research interest are in Digital Image Processing, Computer Vision and Intelligent System. Currently, she is the staff of Department of Informatics, Faculty of Engineering, Hasanuddin University. She is a member of IAENG and member of IEEE.