Mitigating Bias and Assessment Inconsistencies with BERT-Based Automated Short Answer Grading for the Indonesian Language

Amalia Amalia, Maya Silvi Lydia, Muhammad Anggia Muchtar,
Fuzy Yustika Manik, Sinu, and Dani Gunawan

***Abstract*—Automating the grading of short answers in Indonesian presents unique challenges, primarily due to the inherent variability in student responses and the limited linguistic resources available for fine-tuning models. This study aims to develop an automated grading system for short answers by employing a modified BERT model tailored explicitly for the Indonesian language. Our methodology involves translating the Stanford Question Answering Dataset (SQuAD 2.0) and augmenting it with domain-specific content from lecture materials, resulting in an additional 1,000 question-answer pairs. Furthermore, we optimize the model using the Optuna library for hyperparameter tuning. Experimental results show a minimum loss of 1.81 for the model, with optimized hyperparameters including a learning rate of 3.36 × 10⁻⁵, a training batch size of 8 per device, and two training epochs. The model achieves an F1-score of 69%, demonstrating a satisfactory level of accuracy comparable to or slightly exceeding typical performances on SQuAD 2.0 in English, which averages around 66%. Evaluations were conducted across various scenarios, including short-answer, long-context, and long-answer assessments. The results revealed that short answers achieved the highest cosine similarity and a perfect QWK score of 1. In contrast, long-context and long-answer scenarios yielded a QWK of 0.5, indicating a need for further improvement; however, the model shows promise for long answers with additional enhancements. This study demonstrates BERT's potential to enhance equity and precision in grading Indonesian short answers, despite limitations in response completeness.**

***Index Terms*— Indonesian Short Answer Grading, Automated Grading System, Automated Essay Scoring, BERT**

Manuscript received March 11, 2024; revised January 8, 2025 This work was supported by the Ministry of Research and Technology and Higher Education, Republic of Indonesia, under a grant for Research Fundamental of Year 2023, in accordance with the Letter of Assignment Agreement, Number: 167/E5/PG.02.00.PL/2023.

Amalia Amalia is an associate professor in the Department of Computer Science, Universitas Sumatera Utara, Medan, Indonesia (Corresponding Author to provide phone: +62811645554; e-mail: amalia@usu.ac.id).

Maya Silvi Lydia is an associate professor in the Department of Computer Science, Universitas Sumatera Utara, Medan, Indonesia (e-mail: maya.silvi@usu.ac.id).

Muhammad Anggia Muchtar is an assistant professor in the Department of Information Technology, Universitas Sumatera Utara, Medan, Indonesia (e-mail: anggi.muchtar@usu.ac.id).

Fuzy Yustika Manik is an assistant professor in the Department of Computer Science, Universitas Sumatera Utara, Medan, Indonesia (e-mail: fuzy.yustika@usu.ac.id).

Sinu is an undergraduate student in the Department of Computer Science, Universitas Sumatera Utara, Medan, Indonesia (email: sinu@students.usu.ac.id).

Dani Gunawan is an assistant professor in the Department of Information Technology, Universitas Sumatera Utara, Medan, Indonesia (e-mail: danigunawan@usu.ac.id).

I. Introduction

FORMAL educational environments, such as schools and universities, employ various assessment methods, including multiple-choice tests, essays, project-based tasks, and case-study analyses. These assessments can generally be categorized into two main types: those designed to evaluate higher-order thinking skills (HOTS) and those targeting lower-order thinking skills (LOTS).

LOTS assessments primarily evaluate students' ability to recall information, take notes, replicate processes, and follow instructions. In contrast, HOTS questions are designed to cultivate advanced cognitive skills, challenging learners with complex tasks that require analytical thinking and creativity [1].

In educational assessment, LOTS are often evaluated using multiple-choice formats, which are well-suited for automated grading and assessing fundamental understanding. In contrast, essay assessments are commonly employed to foster HOTS, as they encourage analytical and creative thinking. However, unlike LOTS formats, such as questions in multiple-choice or true/false formats, HOTS-focused assessments, particularly essays, present significant challenges for automated evaluation.

In general, there are two categories of essay questions: long-answer and short-answer. Long-answer questions assess various components, including spelling, grammar, sentence coherence, and alignment with the main topic [2]. Meanwhile, short-answer questions focus on assessing a student's understanding of specific concepts. Short Answer Grading (SAG) emphasizes evaluating concise responses, typically 1 to 3 sentences, by comparing them to a model answer. While grammar and coherence are considered in SAG, they are generally less critical to the overall assessment [3].

While coherence is often less emphasized, short-answer questions remain challenging due to the variability in students’ responses. Students may provide different answers while conveying similar meanings and intentions. Furthermore, the manual grading of short-answer questions, especially when involving multiple graders, is prone to bias, human error, and significant time demands. Consequently, an automated scoring system is necessary to ensure impartiality and efficiency [2]. Usually, two main scoring techniques are used: instance-based and similarity-based methods. Similarity-based techniques assign scores by comparing students' responses to standard answers.

Conversely, instance-based methodologies involve training a model to identify and differentiate the characteristics of responses corresponding to various scoring levels [4]. The selection of techniques for content evaluation warrants careful consideration by researchers. Earlier research has thoroughly investigated Short Answer Grading (SAG), with several studies utilizing traditional feature extraction techniques like n-gram language models. On the other hand, certain approaches have utilized deep learning techniques like BERT (Bidirectional Encoder Representations from Transformers) to enable automated feature extraction. According to several studies, implementing BERT has demonstrated superior accuracy [5]. BERT utilizes a dual approach, encompassing the direct application of pre-trained models as well as the fine-tuning of these models in combination with classification algorithms specifically tailored for Short Answer Grading (SAG) tasks. Fine-tuning BERT for automated essay scoring frequently utilizes datasets tailored for question-answering tasks, such as the Automated Student Assessment Prize (ASAP) dataset, which is accessible on Kaggle (https://www.kaggle.com/c/asap-aes). This dataset contains student responses and reference answers. However, the ASAP dataset does not explicitly model the relationship between the prompt and the student’s response. As a result, models trained on ASAP may predict overall quality but may not effectively capture how well the content aligns with the specific requirements of the question or prompt. Consequently, the ASAP dataset is less appropriate for models that prioritize assessing the relevance between a student's response and the corresponding essay prompt. The Stanford Question Answering Dataset (SQuAD) is another commonly used resource for question-answering tasks. A significant challenge for non-English languages is the lack of available linguistic resources and datasets [6]. Therefore, the SQuAD dataset has been adapted into multiple languages, including Spanish [7], Persian [8], Dutch [9], Bengali [10], and others. However, since SQuAD is designed for question answering rather than grading, additional steps, such as calculating answer similarity, are required when applying it to SAG.

This study addresses the gap in automated grading resources for the Indonesian language by developing a SAG system. Through translation and enrichment procedures, the system modifies the SQuAD dataset and applies fine-tuning approaches to a pre-trained BERT model. It enables automated grading of student responses based on educator-provided questions and accompanying text passages. This study is one of the first to develop an automated SAG system for Indonesian, making a significant contribution by addressing the scarcity of resources and offering an efficient solution for mass grading. The framework of this study includes a review of related literature to highlight relevant prior research, an examination of the application of BERT for Short Answer Scoring, and a detailed discussion of the methodology. The results are then presented and analyzed, followed by the conclusion.

II. Related Works

Early research in Automated Essay Scoring (AES), also known as Automated Essay Grading (AEG), commenced with the inception of an application known as Project Essay Grade (PEG) in 1966 [11]. PEG relies on fundamental linguistic features such as sentence length, word frequency, and word density. Essays with longer sentences are attributed higher scores under this framework, under the presumption that students capable of composing lengthier essays possess proficient language skills and the capacity to articulate ideas coherently. The evolution of AES applications continued until 1996 [12]. Subsequently, in 2003, a novel approach to AES emerged, titled the Intelligent Essay Assessor (IEA), developed by [13]. The IEA approach introduced the evaluation of essays based on coherence, encompassing an assessment of the relationships between sentences and paragraphs. Among the prominent algorithms within IEA is latent semantic analysis (LSA), devised to deduce an essay's underlying meaning, even in cases where the wording deviates from the reference answer [14]. Numerous scholars have explored the development of AES utilizing the LSA approach, including studies by [15][16][17][18]. These studies conducted AES research employing LSA specifically for the Indonesian language. While LSA research has yielded promising outcomes, the algorithm still encounters challenges when confronted with discrepancies in the lengths of student and lecturer responses, potentially affecting result accuracy.

In response to these challenges, researchers have increasingly gravitated towards leveraging deep learning techniques for automated AES [19]. For example, [17] integrated Word2Vec as a word embedding method in a document retrieval task employing queries. Their investigation yielded an accuracy rate of 85% for large datasets and 52.5% for smaller datasets, although Word2Vec's efficacy is constrained, notably in handling Out of Vocabulary (OOV) terms. Furthermore, [20] explored deep learning-based AES research by integrating the computation of string similarity in student responses with word embedding techniques. Prior research indicates that deep learning methodologies epitomize an advanced approach for AES, particularly within English language contexts [19]. The effectiveness of deep learning methodologies in AES is intricately linked to the accessibility of extensive language learning datasets. A relatively novel area of AES investigation integrates components of assessment instruments or rubrics. For example, [21] utilized a rubric comprising diverse assessment criteria such as content, sentence structure, evidence, writing style, and skills. Various algorithms were applied for each evaluation criterion, including a multiple regression approach for assessing writing style and a cosine similarity algorithm for content evaluation. Advancements in AES facilitated by the rubric approach have also garnered attention from other researchers [22][23]. Additional studies about automated grading, such as the work by [24], have been conducted. In another AES investigation, [25] employed LSTM (Long Short-Term Memory) and RNN (Recurrent Neural Network) algorithms. It was observed that LSTM and RNN models exhibit restricted memory capacity when tasked with processing lengthy textual content or knowledge in question-answering contexts. The recent adoption of BERT for AES, including Short Answer Grading (SAG), has demonstrated promising advancements.

A study by [26] underscored BERT's principal strength, residing in its self-attention mechanism, which considers both preceding and subsequent contexts. This capability enables BERT to anticipate words and comprehend sentences, rendering it suitable for question-answering tasks.

Moreover, additional investigations corroborate BERT's efficacy in question-answering tasks, such as the study conducted by [27], which effectively implemented BERT for question answering, particularly within the BioSQ dataset, achieving an F1-score of 76.44%. Sung et al. [28] demonstrated that the performance of the pre-trained BERT model can be enhanced by integrating data derived from domain-specific resources, such as textbooks. Short answer evaluations perform better when an upgraded pre-trained language model is tailored for certain tasks, according to empirical studies on multi-domain datasets. Several prior investigations have concentrated on examining approaches for comparing student answers with reference answers, exemplified by the study conducted by [4]. This research constitutes a comprehensive review of various methodologies about similarity metrics, encompassing techniques such as the computation of syntactic and semantic similarity measures and sentence embedding. According to findings from [4], sentence embedding utilizing BERT emerges as a robust methodology. This assertion is supported by a study conducted by [29], which demonstrated that sentence embeddings derived from BERT can serve as a standalone semantic feature, enabling direct comparisons using basic methods such as cosine similarity. Additionally, previous research on scoring evaluation methodologies has been carried out by [30].

Many researchers have conducted relevant studies on suitable datasets for building question-answering models in various languages. One of the most widely adapted datasets in multiple languages is the SQuAD dataset. Question-answering models have been developed for languages including Spanish [7], Persian [8], Dutch [9], and Bengali [10], demonstrating that the SQuAD dataset can be translated into different languages and has shown good performance. However, most do not surpass the English SQuAD baseline.

Several prior studies focused on the Indonesian language have utilized the SQuAD dataset, including [6][31] and [32]. However, these studies concentrated on developing models that automatically generated questions in Indonesian rather than addressing question-answering scoring. Relevant research on automated SAG in the Indonesian language was conducted by [33] and [34]. These studies used self-constructed datasets with a format similar to the ASAP dataset, consisting of reference answers, student responses, and holistic scores. Both studies explored automated SAG for the Indonesian language using BERT. For instance, the study by [33] created a dataset of 36 questions and 9,165 answers from 534 respondents in Biology and Geography, graded by seven experts. It compared various pre-trained models, including Word2Vec and BERT variants, finding that BERT outperformed Word2Vec.

Building on insights from prior research, this study aims to develop a short-answer grading model specifically designed for the Indonesian language using an advanced computational framework. To optimize performance, hyperparameters were carefully tuned through extensive experimentation. Additionally, the dataset was enriched with domain-specific resources sourced from lecture materials. The model processes a question and reference text to generate an authoritative response, then compared to student submissions to calculate student scores. This system is designed to enable efficient and accurate scoring for entire classes.

III. BERT for Short Answer Task

BERT is a language model built upon the Transformer architecture [5]. A key element of the Transformer framework is the Attention Mechanism [35], which enables the model to concentrate on critical information within the context of a sentence. BERT encompasses at least two architectural variants: $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$.

The main difference is in the size parameters; $\text{BERT}_{\text{BASE}}$ has 110 million parameters, 12 levels, a hidden size of 768, and 12 Self-Attention Heads. In contrast, $\text{BERT}_{\text{LARGE}}$ has 340 million parameters overall, 24 layers, a hidden size of 1024, and 16 attention heads. The choice of architecture depends on task-specific requirements; $\text{BERT}_{\text{LARGE}}$ exhibits enhanced capability in understanding intricate contexts but necessitates more substantial memory and computational resources compared to $\text{BERT}_{\text{BASE}}$.

Consequently, in resource-constrained environments, $\text{BERT}_{\text{BASE}}$ may represent a more pragmatic selection. BERT, representing a state-of-the-art approach in language modeling, manifests effective performance within the domain of SAG. Diverse methodologies exist for leveraging BERT to tackle SAG challenges, including using pre-trained BERT models alongside classification algorithms or fine-tuning pre-existing BERT models to align with the specific demands of SAG tasks. However, several studies suggest that fine-tuning for grading categories does not consistently outperform standard classification techniques in addressing essay grading problems encompassing short essay grading scenarios. Moreover, the process of fine-tuning entails higher computational costs [36]. A feasible strategy for leveraging pre-trained BERT in question answering involves its training on specific datasets, such as SQuAD. SQuAD is widely used for developing question-answering models due to its structured format and extensive data collection. It is a valuable tool for training and assessing different natural language processing models. It comprises pairs of questions and answers extracted from Wikipedia articles. Fine-tuning using the SQuAD dataset specifying the starting and ending positions of words corresponding to the correct answers within the passage text—the fine-tuning process utilizing the SQuAD dataset is depicted in Figure 1.

The sentence pairs are concatenated into a unified sequence comprising a question and corresponding reference text. Two distinct methods are employed to differentiate between sentences. First, a special token ([SEP]) is inserted to separate them. Second, a specific embedding is assigned to each token, indicating its association with either sentence A or B. Additionally, each sequence begins with a classification token ([CLS]), whose resultant hidden state serves as the overall sequence representation for classification. During the fine-tuning stage, a starting vector $S \in \mathbb{R}^{H}$ and an ending vector $E \in \mathbb{R}^{H}$ are introduced specifically for this purpose.

The probability P_i of word i representing the start of an answer span is calculated by performing a dot product between T_i and S, followed by applying a softmax function across all words in the paragraph, as shown in Equation 1.

$$P_i = \frac{e^{S.T_i}}{\sum je^{S.T_i}} \tag{1}$$

The token with the highest probability of being the starting token is selected. A similar process is used to determine the concluding token, utilizing a distinct weight vector designed for this task.

The SQuAD dataset was initially developed to train question-and-answer systems in English. Subsequent measures are necessary to adapt it for use in Indonesian. This study incorporated a dataset specifically designed for computer science instructional content. The methodology section provides a detailed explanation of these adaptation measures.

IV. Methodology

BERT is considered one of the leading models for addressing a wide range of complex tasks in Natural Language Processing (NLP), including question-answering task. This study aims to adapt the model for the Indonesian language context, requiring the integration of supplementary methodologies. To address potential errors and mitigate assessment biases, the question-answering model is designed to generate reference responses based on queries and instructional materials provided by instructors. An automated scoring mechanism is applied, leveraging cosine similarity to compare reference responses with student answers. The resulting application enables the automatic computation of student scores within a single classroom setting.

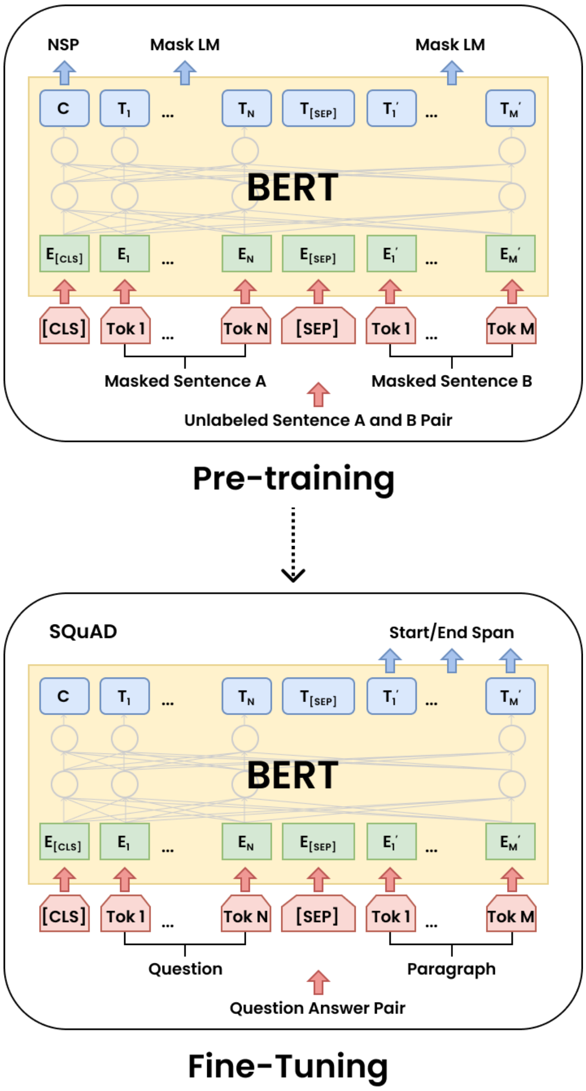

Fig 1. Overall pre-training and fine-tuning procedures for BERT Question Answering [5]

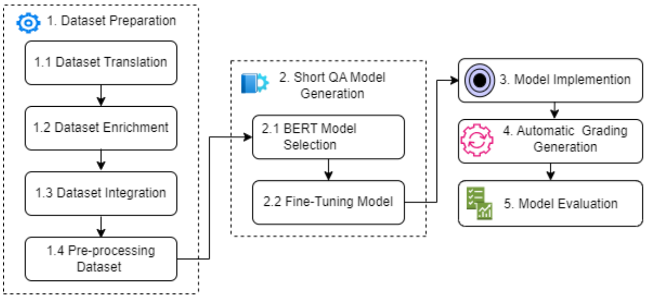

Fig 2. The overall methodology

The study followed a five-stage process: dataset preparation, development of a short question-answering model, model implementation, automated grading generation, and model evaluation. Figure 2 provides an overview of the entire process.

A. *Dataset Preparation*

Dataset preparation is crucial, as a model requires a high-quality dataset to generate accurate reference answers. Understanding the characteristics of the original dataset is essential. SQuAD version 1.1 consists of over 107,785 questions created by crowd workers from Wikipedia articles, with answers provided as text excerpts extracted from the corresponding passages [37]. However, SQuAD 1.1 includes only questions with clear answers in the text, meaning that models trained on this dataset may struggle to handle scenarios with no answer. Additionally, the limited variety in question types leads to models focusing on specific patterns for answer recognition, reducing their adaptability to diverse question forms [38].

SQuAD 2.0 was introduced to address these limitations by adding 53,775 unanswered questions. This version enables models to be trained for greater flexibility and enhances their ability to assess answer relevance across various contexts. The unanswered questions were generated by introducing antonyms, modifying numerical values, and applying negation to questions with known answers [39]. In this study, we generated ID-SQuAD 2.0, an adaptation of the SQuAD 2.0 dataset. We chose SQuAD 2.0 as the baseline because it better prepares models for real-world question-answering scenarios, where not all questions have clear or direct answers. To better accommodate the characteristics of the Indonesian language, we implemented additional steps in ID-SQuAD 2.0: dataset translation, enrichment, integration, and pre-processing.

Dataset Translation

We executed the translation process utilizing the Google Neural Machine Translation System API [40], followed by manual validation conducted by human annotators. This validation stage is pivotal due to the potential inaccuracies in the translation results. Furthermore, manual verification ensures that the conformity of the starting and ending points of tokens used for answer identification within the translated dataset conforms to the conventions of the Indonesian language.

Dataset Enrichment

The SQuAD dataset offers a broad range of topics and contexts, serving as a solid foundation for Indonesian SAG models. We enriched the dataset with computer science course materials to tailor it to specific fields. We involved ten educators to validate and refine additional question-answer pairs for this enrichment. Although the manual creation of these pairs is ideal, it could be more practical for large datasets. Therefore, we created a natural language processing tool to automate the creation of question-answer pairs from text segments in educational materials. This utility creates factoid questions using a template-based approach, which generates questions based on predefined templates filled with specific information from the text. Our choice of factoid questions aligns with the SQuAD dataset's focus on extracting specific factual information. The flowchart of this process is shown in Figure 3.

1) Input Text Span

The text span is the primary input for the question-answer pair generation process. This study's text span consists of excerpts or sections of teaching materials input by validating lecturers. The utility accepts teaching materials in various formats (PDF, TXT, PPT) converted into TXT for processing.

2) Pre-processing

Pre-processing includes tokenization, normalization, and removing irrelevant elements to transform the text into an organized format for further analysis. This step aims to prepare and analyze the text span to extract relevant elements that will facilitate the creation of meaningful and accurate questions. Pre-processing involves preparing the text for subsequent analysis by performing tasks such as tokenization, normalization, and the removal of irrelevant elements. The objective is to transform the text into a structured format that facilitates efficient analysis.

3) Identifying Key Information

Identifying key information involves extracting essential components from the pre-processed text, such as key phrases, entities, and relationships. Key elements are identified using Part-Of-Speech (POS) tagging, which helps recognize nouns, verbs, or adjectives. We implemented POS tagging for the Indonesian language based on previous research by [41]. Named Entity Recognition (NER) is used to identify essential entities like names, organizations, and locations, playing a crucial role in structuring textual information [42]. We implemented NER for the Indonesian language based on previous research by [43].

4) Identify Phrase Patterns

After pre-processing, the next step is to identify phrases and their patterns. Phrases are grammatical units composed of two or more words that function together. By recognizing these grammatical functions, the system can generate accurate and contextually relevant questions to extract specific information from the text. In this study, we based our identification of phrase patterns on previous research by [44].

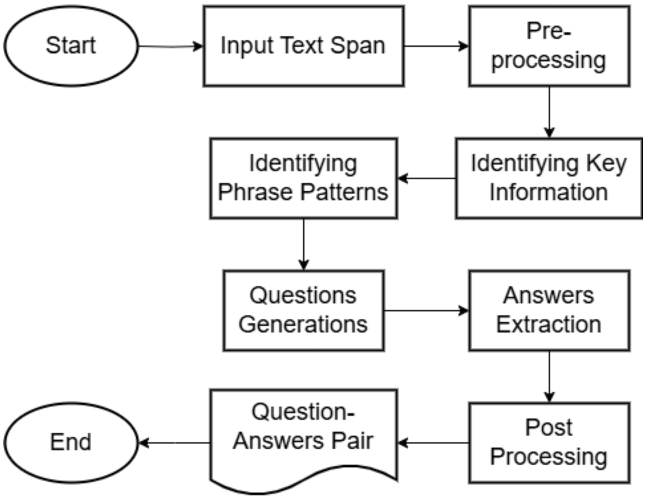

Fig 3. Dataset Enrichment Flowchart

5) Question Generation

The next step is question generation; these questions are created using pattern matches informed by keywords and grammatical functions identified in prior studies [44].

6) Answer Extraction

Answer extraction is performed through span identification based on pattern matches from the previous step. The application determines the beginning and end positions of the answer within the text.

7) Post-Processing

Post-processing involves refining and finalizing the generated question-answer pairs to ensure accuracy, clarity, and relevance. Volunteer lecturers validate the questions for grammatical correctness and coherence, filter out nonsensical or irrelevant questions, refine them for clarity and precision, ensure correct alignment of corresponding answers, and format the questions and answers according to the desired output, such as the SQuAD dataset format, to ensure consistency and usability for model training. Using this iterative approach, we generated 1,000 question sets based on 100 paragraphs, with each set including tokens that mark the beginning and ending positions of the answer within the text.

Dataset Integration

The dataset integration process aims to consolidate the translated SQuAD dataset with information extracted from instructional materials into a cohesive dataset. At this stage, we standardize the structure of the integrated dataset to align with SQuAD dataset format, including fields such as *field_id, title, context, question,* and *answers*. The *field_id* and *title* fields correspond to the identification number and title of the context, respectively. The *question* field contains the posed inquiry, while the *context* field represents the passage text that may contain potential answers. In the *answers* field, data is structured as a dictionary containing *text* and an *answer_start* attribute. The *text* component represents the answer to the posed question within the *context*, while *answer_start* indicates the index within the *context* where the model-identified correct answer begins. These fields also show whether an answer is present or absent within the context.

Pre-Processing Dataset

Initial pre-processing is imperative to ensure the model can process the acquired dataset efficiently. This involves transforming the dataset into a compatible input format for the model, involving delineating questions and passage texts using specialized tokens like [CLS] at the beginning and [SEP] to separate the question from the passage. The dataset pre-processing involves two main phases: tokenization and input embedding. This study uses the Hugging Face platform [45], which offers comprehensive libraries for developing transformer-based models, including BERT [46]. Tokenization breaks text into smaller units, enabling BERT to understand context effectively. We use the BERT AutoTokenizer library to tokenize questions and passages from the dataset, ensuring efficient processing. This library also removes extraneous details, such as punctuation and symbols, from the input.

B. Short Question-Answer Model Generation

This phase involves constructing a concise question-answering model tailored to the Indonesian language. The process consists of two primary stages: selecting a pre-trained BERT model and tailoring it to the specific task through fine-tuning.

Pre-Trained BERT Model Selection

A pre-trained BERT model is a deep learning framework constructed using the transformer architecture, trained on an extensive corpus of text through unsupervised learning techniques. These models encapsulate broad language understanding derived from extensive datasets, making them highly effective for various NLP applications. This study utilized a pre-trained BERT model named IndoBERT [47], explicitly employing the IndoLEM/IndoBERT base-uncased variant. specifically the IndoLEM/IndoBERT base-uncased variant. IndoBERT serves as a foundational benchmark for vector representation in the Indonesian language, having been rigorously trained on a comprehensive corpus of over 220 million words from diverse Indonesian sources. Its selection was based on its demonstrated accuracy and effectiveness in numerous prior studies within the Indonesian language domain, affirming its suitability for our research objectives.

Fine Tuning Model

The fine-tuning phase is aimed at adapting the pre-trained model to accurately comprehend and respond to questions in Indonesian. This process significantly improves the model's ability to generate precise predictions for queries in the Indonesian language. The model is designed to predict responses to user-generated questions. The training process involves iterative steps to determine the presence of an answer within the provided context. If no answer is identified, the system generates a warning message.

Conversely, if an answer is found, the process identifies the *answer_start* and *answer_end* positions within the context. To enhance the accuracy of these predictions, the model is initialized with randomly assigned weights. Following this, the process involves calculating the loss function, which measures the extent to which the model's predictions align with the true values. Key hyperparameters optimized during training include the learning rate, number of epochs, weight decay, and batch size for both training and development datasets.

The hyperparameter exploration process is executed by the predefined *max_evals* parameter. In this investigation, the *max_evals* value was established at three due to constraints in computational resources. Following the completion of the hyperparameter exploration phase, the hyperparameters associated with the lowest loss are identified as the most optimal configuration for the model. This study employed the Optuna library [52], which employs the Tree-structured Parzen Estimator (TPE) method for hyperparameter exploration. Various hyperparameters investigated include *learning_rate*, *per_device_train_batch*, *per_device_eval_batch*, *num_train_epochs*, and *weight decay*. Subsequently, the trained model was integrated into the HuggingFace Library [45] [53], enhancing its overall accessibility. To illustrate the procedural steps, we present the pre-processing process in Figure 4.

We utilized the BERT base model configuration to transform token representations into embeddings, which support a maximum input length of 512 tokens. To handle inputs exceeding this limit, the texts were divided into smaller, manageable segments, referred to as “chunks”. A “document stride” variable was introduced to regulate the overlap between chunks and ensure efficient processing. Our analysis determined an optimal token allocation of 384 tokens per chunk with a document stride of 128 tokens, resulting in the document being divided into three distinct chunks, each containing 128 tokens. This configuration prevents excessive strain on the model's processing

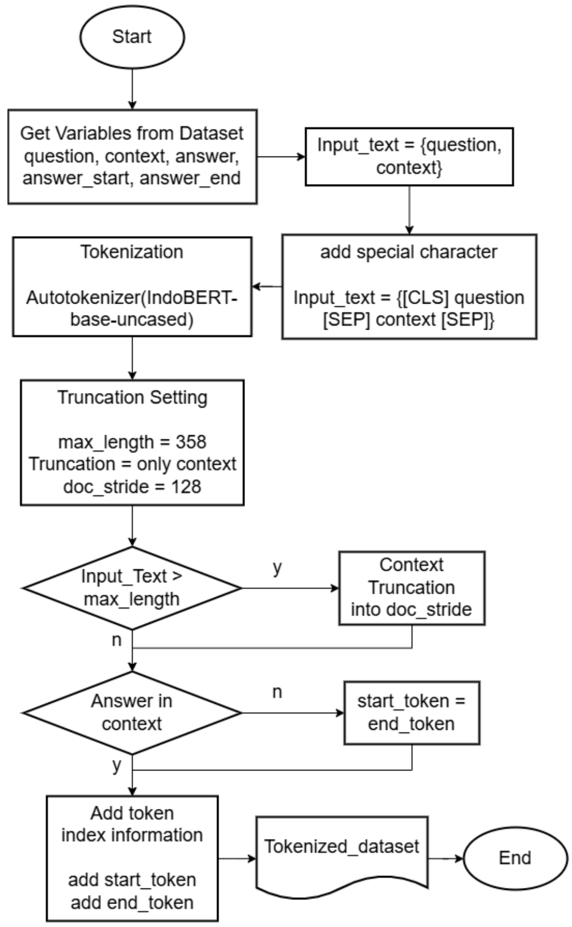
Start
Get Variables from Dataset question, context, answer, answer_start, answer_end
Input_text = {question, context}
add special character
Input_text = {[CLS] question [SEP] context [SEP]}
Tokenization
Autotokenizer(IndoBERT-base-uncased)
Truncation Setting
max_length = 358
Truncation = only context
doc_stride = 128
Input_Text > max_length
y
n
Context Truncation into doc_stride
Answer in context
n
y
start_token = end_token
Add token index information
add start_token
add end_token
Tokenized_dataset
End

Fig 4. Pre-processing flowchart

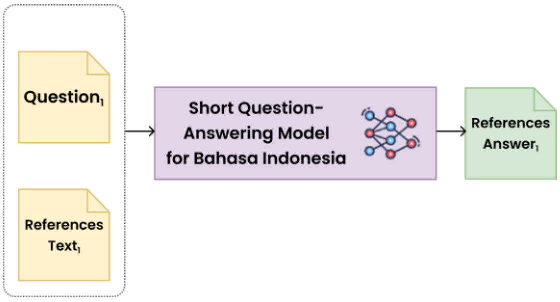

Fig 5. Model Implementation of Short Question-Answering BERT Model for the Indonesian language

capabilities and allows each chunk to be independently processed by the BERT model. The outputs from these chunks are merged to generate the final result.

Employing a document stride mechanism augments the efficiency and effectiveness of the BERT model in handling lengthier documents. This enhancement is realized through a deliberate reduction in the simultaneous processing of tokens, optimizing the utilization of computational resources and facilitating the seamless incorporation of document context within the model's analytical framework.

C. Model Implementation

The proposed model generates answers to user-provided questions by extracting relevant information from the reference text. To manage multiple inquiries efficiently, all responses are systematically stored in a database, indexed by their corresponding question numbers. Figure 5 shows a visual representation of this process.

D. Automatic Short Answer Grading

In the grading phase, a comparative analysis is conducted between the student responses and the reference answers generated in the previous stage, adhering to the methodology described in [29]. Both student and reference answers are converted into BERT representations, from which sentence embedding values are computed. This computation involves summing the vector values of all tokens within each answer, a technique known as the Sum of Word Embeddings (SOWE), as shown in Equation 2.

$$a_{ij} = \sum_{j=1}^{n_j} W_k \tag{2}$$

In this context, a_{ij} denotes the j^{th} answer vector for question q_i, and W_k represents the vector corresponding to the k^{th} word in the answer a_{ij}.

The similarity between each student's answer a_{ij} and the desired answer a_i is measured using cosine similarity, as given in Equation 3.

$$cos(a_{ij}, a_i) = \frac{a_{ij}.a_i}{|a_{ij}||a_i|} \tag{3}$$

The similarity scale was adjusted to range from 0 to 10. Subsequently, the final score is computed considering the weight assigned to the question by the instructor. When the instructor provides no specific weight, the question's weight is determined as the average among all questions. The computation of the final score is given in Equation 4.

$$FinalScore = \sum_{i=1}^{Q} cos_i.W_{qi} \tag{4}$$

Where *Cos*$_{\text{i}}$ represents the cosine similarity result for the student's answer to Q questions provided by the instructor, *Wqi* represents the weight assigned to each question by the instructor. Figure 6 illustrates the grading phase.

E. Model Evaluation

Model evaluation is essential for assessing the proposed models' performance, effectiveness, and accuracy. This study developed two models: a Short Question-Answering (QA) model and a Short Answer Grading (SAG) model. The performance of the Short Question-Answering model was assessed using F1 and Exact Match (EM) metrics, which serve to evaluate the accuracy with which the model generates reference answers in comparison to the ground truth provided in the dataset.

In parallel, the Short Answer Grading model was assessed using Cosine Similarity and Quadratic Weighted Kappa (QWK) [49] to measure its performance. Cosine Similarity quantifies the semantic alignment between two text vectors. In the context of Short Answer Grading (SAG), it is utilized to evaluate the degree to which a student's response aligns with the reference answer. Capturing semantic equivalence allows for evaluating answers that may be phrased differently but convey similar meanings.

On the other hand, QWK is a standard metric in essay grading tasks [50] [51], commonly used to measure the agreement between scores from a model and those from a human rater. This evaluation helps determine how closely the model's scores align with human ratings, with higher values indicating more consistent, human-like assessments. In this study, QWK was employed to assess the model's capability to categorize answers into predefined classes, helping to determine whether it mirrors human judgment and minimizes bias.

V. Results and Discussion

A. Short Question-Answering Model Evaluation

The short question-answering model's performance is influenced by several factors, with the dataset's quality being paramount. In this study, we utilized the SQuAD dataset, focusing on the quality of translation and data enrichment to create a high-quality Indonesian version, ID-SQuAD 2.0. Table 1 compares the dataset statistics between the translated ID-SQuAD 2.0 and the original English SQuAD.

The next step is identifying the most effective hyperparameter values to achieve heightened accuracy in generating reference answers based on the input text span and question. This exploration employs the Optuna library [52] to determine optimal hyperparameter configurations across three experimental runs. The hyperparameter values derived from these trials are detailed in Table 2. The third iteration yielded the lowest loss, marking it at 1.81. Consequently, the model architecture was developed based on the hyperparameter settings discerned during the third experimental trial.

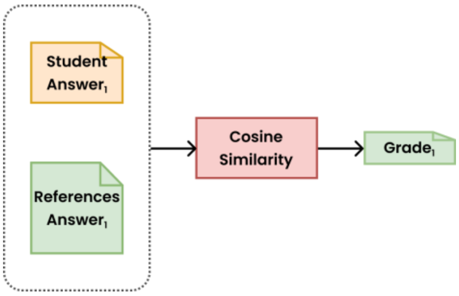

Fig 6. Automatic Short Question Answering Grading

TABLE I DATASET STATISTICS OF ID-SQUAD 2.0, COMPARED TO THE BASELINE SQUAD 1.1 AND SQUAD 2.0

Dataset	Language	Total Questions	Total Articles	Negative Examples
SQuAD 1.1	English	107.702	536	0
SQuAD 2.0	English	151.054	505	53.775
Direct Translated SQuAD 1.1	Indonesian	107.702	536	0
Direct Translated SQuAD 2.0	Indonesian	151.054	505	53.775
ID-SQuAD 2.0	Indonesian	152.054	605	53.775

TABLE II HYPERPARAMETER CONFIGURATION EXPERIMENTS

Loss	Learning Rate	Train Batch	Eval Batch	Epoch	Weight Decay
2.05	1.25 x 10^{-6}	8	16	3	1.96 x 10^{-8}
1.83	1.00 x 10^{-6}	16	16	3	5.85 x 10^{-8}
1.81	3.36 x 10^{-6}	8	8	2	2.27 x 10^{-8}

TABLE III EXACT MATCH (EM) AND F1 SCORES COMPARISON WITH THE BASELINE MODEL

Dataset	Model	EM	F1-Score
SQuAD 2.0	DocQA + Elmo	63.4	66
Direct Translated SQuAD 2.0	IndoBERT	61.4	65.1
ID-SQuAD 2.0	IndoBERT	66	69

TABLE IV COMPARISON RESULTS IN TERMS OF F1 AND EM SCORES ON THE ID-SQUAD 2.0 USING VAROIUS REPRESENTATIONS

Model Representation	EM	F1-Score
LSA with TF-IDF	19	28
Word2vec	29	40
IndoBERT (hyperparameter default from hugging face)	40	30
IndoBERT + our hyperparameter configuration	66	69

Based on these hyperparameters, we fine-tuned the dataset using IndoBERT. The evaluation was performed using the get_eval_dataloader trainer library from Hugging Face [53], and performance was measured using the SQuAD metrics: F1 score and Exact Match (EM). The dataset was split, with 80% designated for training and 20% for testing.

TABLE V QUESTIONS SET FOR EVALUATION

Type of Question Set	Number of Questions	Description
Easy Short Answer (ESA)	20	Questions with short answer of 1 to 5 words based on a context of approximately 100 to 300 words.
Long Context (LC)	20	Questions based on a long context of 800 words or more.
No Answer (NA)	20	Questions that contain no answers in the context/text span.
Long Answer (LA)	20	Questions that contain long answers, more than 10 words.
Specific Domain (SD)	20	Questions from specific domains that have not been previously included in existing datasets.

We compared our results with baseline models for SQuAD 1.1 and SQuAD 2.0 in English, using pre-trained DocQA models with ELMo [38]. The comparison details are provided in Table 3. The model's performance yielded an F1-score of 69% and an EM of 66%, indicating impressive results. This accomplishment is significant, as it aligns with or slightly surpasses the anticipated performance of models assessed on SQuAD 2.0 in English. According to our assumption, this success is attributed to the enrichment process and the smaller number of negative examples compared to the baseline model for English. The dataset with direct translations from SQuAD 2.0 resulted in lower F1 and EM scores compared to the baseline English SQuAD 2.0. These results suggest that the model trained on the ID-SQuAD 2.0 dataset demonstrates superior quality, allowing it to achieve state-of-the-art performance in question-answering tasks for the Indonesian language. During the evaluation phase, we conducted a comparison between our question-answering model and earlier approaches, such as Latent Semantic Analysis (LSA), Word2Vec, and IndoBERT, using default parameters from the Hugging Face library. By incorporating several baseline models, we were able to thoroughly evaluate the strengths and weaknesses of our approach. The results indicate that our model surpasses the others, with a detailed comparison presented in Table 4.

B. *Reference's Answer Generation Evaluation*

In short answer question models based on reading comprehension, such as those used in the SQuAD 1.1 dataset, reference answers must always be derived from specific text segments. In contrast, SQuAD 2.0 introduces a new category of questions known as 'plausible but unanswerable questions,' which may appear reasonable but lack corresponding answers in the text. Therefore, during the evaluation process, it is essential to assess the generated reference answers using various parameters, including the presence of unanswerable questions, the length of text spans, and the occurrence of long answers. This comprehensive approach is crucial for ensuring consistency and accuracy in the model's responses, allowing it to align with the information presented in the text. We categorized the questions into several distinct sets to cover diverse scenarios, including questions with simple/easy answers questions requiring long contexts, questions with no answers found in the context, questions with long answers, and questions from specific domains. Details of these question sets and their descriptions can be found in Table 6.

TABLE VI Example Question Sets With Diverse Scenario.

	Indonesian	English
Context	General Electric (GE) memasuki industri manufaktur komputer pada tahun 1950-an. Selama periode tersebut, GE merupakan pengguna komputer nonpemerintah terbesar di dunia, di luar pemerintah federal AS, dan merupakan perusahaan pertama di dunia yang memiliki komputer. Fasilitas manufaktur utama GE, "Appliance Park," menjadi lokasi nonpemerintah pertama yang memiliki komputer. Namun, pada tahun **1970**, GE memutuskan untuk menjual divisi komputernya ke Honeywell, yang menandai keluarnya GE dari sektor manufaktur komputer. Meskipun demikian, GE tetap menjalankan operasi pembagian waktu selama beberapa tahun. Melalui General Electric Information Services (GEIS, sekarang dikenal sebagai **GXS**), GE menjadi penyedia layanan komputer pembagian waktu terkemuka, termasuk platform komputasi daring seperti GEnie..	General Electric (GE) entered the computer manufacturing industry in the 1950s. During that period, GE was the largest non-governmental user of computers globally, outside of the U.S. federal government, and it was the first company in the world to own a computer. GE's primary manufacturing facility, "Appliance Park," became the first non-governmental location to host a computer. However, in **1970**, GE decided to sell its computer division to Honeywell, marking its exit from the computer manufacturing sector. Nevertheless, GE continued its timesharing operations for several years. Through General Electric Information Services (GEIS, now known as **GXS**), GE became a leading provider of timesharing computer services, including online computing platforms such as GEnie.
Question	Divisi GE mana yang menyediakan layanan berbagi pakai komputer?	Which division of GE offers computer timesharing services?
Actual_answer	**GXS**	**GXS**
Generated_answer	General Electric Information Services	General Electric Information Services
Question	Pada tahun berapa GE menjual divisi komputernya ke Honeywell?	In which year did GE sell its computer division to Honeywell?
Actual_answer	**1970**	**1970**
Generated_answer	1970	1970
Question	Apa bisnis kedua yang memiliki komputer setelah GE?	What was the second business to own a computer after GE?
Actual_answer	<No Answer>	<No Answer>
Generated_answer	Honeywell	Honeywell
Context	Artificial Neural Network (ANN), atau Jaringan Saraf Tiruan (JST), adalah **sistem komputasi yang dimodelkan berdasarkan cara kerja sistem saraf manusia**. Tujuan utama pengembangan Neural Network adalah menciptakan sistem yang mampu belajar secara mandiri berdasarkan data dan kondisi lingkungan yang diberikan**.** Neural Network meniru cara manusia belajar melalui contoh, sebuah proses yang disebut supervised learning. Jaringan ini dikonfigurasi untuk tugas-tugas spesifik seperti pengenalan pola atau klasifikasi data, dan disempurnakan melalui proses pembelajaran iteratif. Proses ini melibatkan **pemberian nilai bobot pada input, membandingkan output yang dihasilkan dengan output yang diharapkan menggunakan fungsi loss, serta menyesuaikan bobot untuk meminimalkan nilai loss**. Setiap neuron memproses input dengan menghitung perkalian dot dengan bobot yang diberikan, menjumlahkannya (weighted sum), menambahkan bias, dan melewatkan hasilnya melalui fungsi aktivasi untuk menghasilkan output akhir dari neuron tersebut.	Artificial Neural Network (ANN), or in Indonesian known as Jaringan Saraf Tiruan (JST), commonly referred to as Neural Network, is **a computational system modeled after the functioning of the human nervous system**. The primary goal of developing a Neural Network is to design a system capable of learning autonomously based on provided data and environmental conditions. Neural Networks mimic how humans learn through examples, a process called supervised learning. These networks are configured for specific tasks such as pattern recognition or data classification and are enhanced through iterative learning processes. **This process assigns weight values to inputs, compares the generated output to the expected result using a loss function, and adjusts the weights to minimize the loss value**. Each neuron processes inputs by calculating the dot product with assigned weights, summing them (weighted sum), adding a bias, and passing the result through an activation function to produce the final neuron output.
Question	Apa itu Artificial Neural Network?	What is an Artificial Neural Network?
Actual_answer	**sistem komputasi yang dimodelkan berdasarkan cara kerja sistem saraf manusia.**	**a computational system modeled after the functioning of the human nervous system.**
Generated_answer	sistem komputasi yang dimodelkan berdasarkan cara kerja sistem saraf manusia.	a computational system modeled after the functioning of the human nervous system
Question	Apa itu proses pembelajaran?	What is the learning process?
Actual_answer	**pemberian nilai bobot pada input, membandingkan output yang dihasilkan dengan output yang diharapkan menggunakan fungsi loss, serta menyesuaikan bobot untuk meminimalkan nilai loss.**	**giving weight values to the input, the output produced is then compared with the output that should be called the loss function value**
Generated_answer	pemberian nilai bobot pada input, membandingkan output yang dihasilkan dengan output yang diharapkan	giving weight values to the input, the output produced is then compared with the output

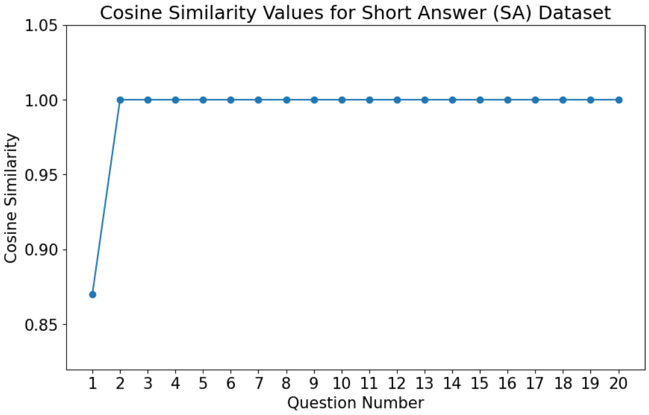

Fig 7. Cosine Similarity Values for Easy Short Answer (ESA) Questions Set

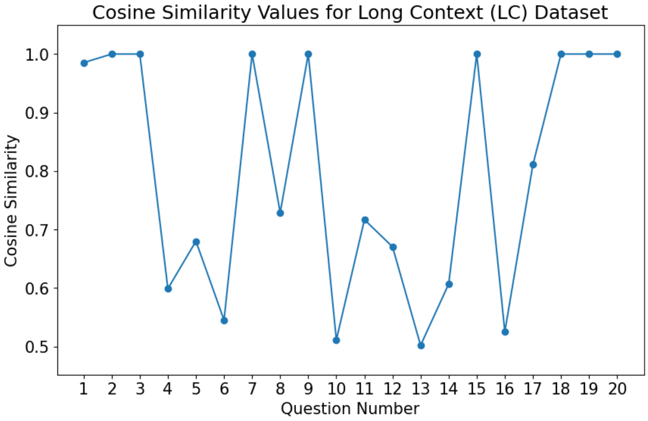

Fig 8. Cosine Similarity Values for Long Context (LC) Questions Set

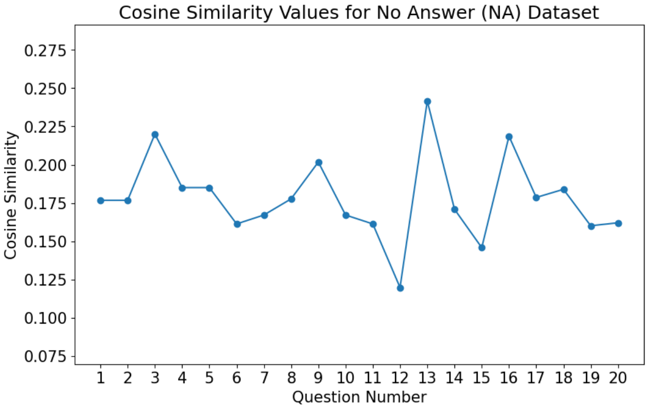

Fig 9. Cosine Similarity Values for No-Answer (NA) Questions Set

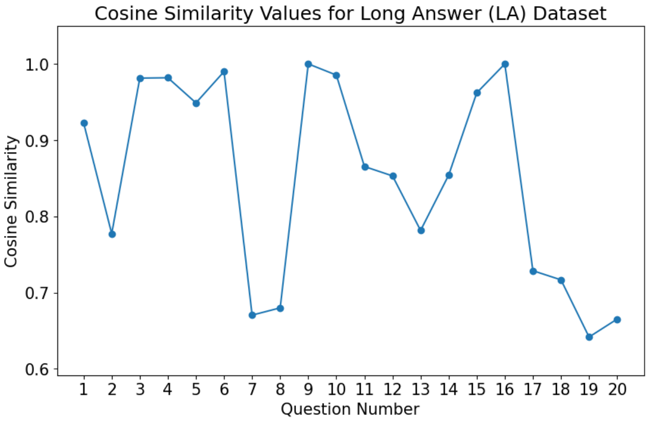

Fig 10. Cosine Similarity Values for Long-Answer (LA) Questions Set

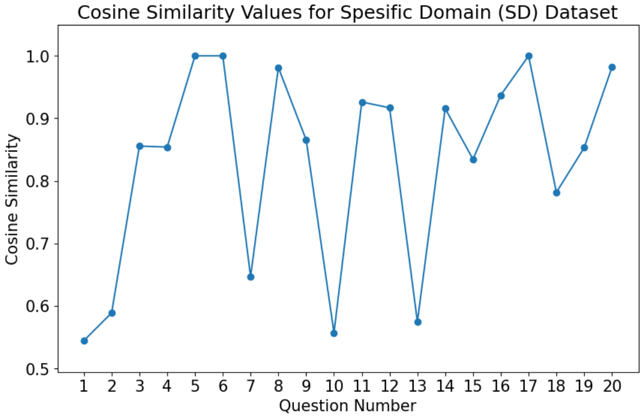

Fig 11. Cosine Similarity Values for Spesific Domain (SD) Questions Set

TABLE VII
QUESTIONS SET AVERAGE COSINE SIMILARITY VALUES

Questions Set	Cosine Similarity Average
Easy Short Answer (ESA)	0.99
Long Context (LC)	0.79
No Answer (NA)	0.18
Long Answer (LA)	0.85
Spesific Domain (SD)	0.83

Figures 7 through 11 present the results of cosine similarity for each question set. A cosine similarity score of 1 indicates complete identity between the two responses, signifying that the model has successfully generated the expected answers. Scores ranging from 0 to 1, with 0 indicating utter dissimilarity, reflect the degree of similarity between the responses. Although the model does not consistently achieve a score of 1, the cosine similarity values do not necessarily indicate incorrect or irrelevant responses. Instead, they suggest that the model produces incomplete answers. The detailed results of the average cosine similarity for each scenario are presented in Table 7. We observe that the Easy Short Answer (ESA) set achieved the highest cosine similarity score of 0.99, indicating the model's successful generation of reference answers when the answers are present in the context and both the context and questions are concise. For the Long Context (LC) and Long Answer (LA) scenarios, the model obtained acceptable cosine similarity scores, ranging from 0.5 to 1, with averages of 0.80 and 0.85, respectively. This suggests that the model can effectively accommodate the partitioning strategy, which involves breaking the input into smaller sections, while still performing well with contexts of 800 words or more. The Specific Domain (SD) question sets achieved an average cosine similarity of 0.83, which is also considered acceptable. The lowest cosine similarity was observed in the No-Answer (NA) question sets, which averaged only 0.17. Based on this evaluation, we conclude that the model is unsuitable for handling No-Answer questions in the context. Figures 7 through 11 show the results of cosine similarity for each question set. A cosine similarity score of 1 indicates complete identity between the two responses, signifying the model's successful generation of the anticipated answers. From 0 to 1, cosine similarity scores 0 denote utter dissimilarity between the responses.

C. Final Score Grading Evaluation

Another aspect of our assessment concerns the grading generated by the system, reflecting the culmination of scores

TABLE VIII. QWK SCORE FOR EACH QUESTION SET SCENARIO

Question Set		**Scoring Classification**																				**QWK**
Easy Short Answer (ESA)	Our Model	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Human Rater	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Long Context (LC)	Our Model	1	1	1	0	2	5	1	2	1	4	2	2	4	2	1	3	1	1	1	1	0.52
	Human Rater	1	1	1	4	5	5	1	2	1	3	5	3	4	5	1	5	1	1	1	1	
Long Answer (LA)	Our Model	1	2	1	1	1	1	2	2	1	1	1	2	3	1	1	1	2	2	2	2	0.42
	Human Rater	2	2	1	1	1	1	3	5	1	1	1	2	2	2	1	1	2	5	4	2	
Spesific Domain (SD)	Our Model	4	3	1	1	1	1	2	1	1	3	3	2	3	1	3	2	1	3	2	1	0.573
	Human Rater	3	4	2	2	1	1	3	1	1	3	1	1	3	1	1	1	1	2	2	1	

obtained from all questions posed. To determine the mean accuracy of the responses generated by our model, Equation addition to accuracy, we also evaluate the grading score from our model using the QWK evaluation. We implemented the QWK metric based on the evaluation results from the question test set in Table 5; however, we excluded the No Answer (NA) question set since we already know that our model is unsuitable for the No Answer scenario. Since QWK is a discrete value for classification, we classified the scores according to classes. In this study, we categorized the scores into five classes of cosine similarity (CS): CS value = 0.80 - 1.00 (Class 1), CS value = 0.60 - 0.79 (Class 2), CS value = 0.40 - 0.59 (Class 3), CS value = 0.20 - 0.39 (Class 4), and CS value = 0.00 - 0.19 (Class 5). The classifications from our model were then compared with human rater scores for each answer in the evaluation question set.

To calculate the QWK value, we used the cohen_kappa_score function with the weights='quadratic' parameter from the scikit-learn library. The QWK results for each question set are provided in Table 8. The highest QWK) score was achieved for the Easy Short Answer (ESA) question set, while the other scenarios scored around 0.5. The average QWK score across all scenarios is 0.62, indicating that our model is reasonably consistent with human raters. However, there is potential for further enhancement to achieve a higher level of agreement [54].

The efficacy of the grading process heavily depends on the model's proficiency in generating a reference answer for each posed question. Instances where the model produces an incomplete response, particularly in cases of shorter answers, significantly influence the cosine similarity scores and QWK scores. This scenario arises because the generated reference answer and the student's response must be complete to achieve a perfect text similarity score. Hence, addressing this issue warrants consideration in future endeavors. Notably, the system effectively assesses questions that do not necessitate extensive responses. This approach underscores the utility of BERT-based evaluation in automated short answer grading in Indonesian, as it aids in mitigating biases and assessment inaccuracies by consistently generating accurate responses despite potential incompleteness in longer answers.

VI. Conclusion

The development of automated short answer grading in Indonesian has been effectively realized through utilizing a BERT model in conjunction with a customized SQuAD dataset. This process entailed various stages, including dataset translation, enrichment by adding 1000 question-passage pairs, and subsequent integration. We employed the pre-trained IndoLEM/IndoBERT-base-uncased model tailored explicitly for the Indonesian language and fine-tuned it using the adapted SQuAD dataset. Hyperparameter optimization was conducted using the Optuna library to adjust model parameters. The results revealed a minimum model loss of 1.81 with optimal hyperparameters are determined as a learning rate of 3.36 x 10^-5, a batch size of 8 per device, and a training duration of two epochs.

Furthermore, the model demonstrated a 69% F1-score, indicative of a commendable level of accuracy, which either matches or slightly surpasses the average performance achieved in English SQuAD 2.0 evaluations, typically around 66%. We conducted evaluations across various scenarios, including (1) easy questions with answers in the context and relatively short responses, (2) long-context questions, (3) questions requiring long answers, and (4) questions with no answer found in the context. The highest cosine similarity and QWK scores were observed in the easy short answer set, achieving a perfect agreement of 1. Long-context and long-answer scenarios produced a QWK of 0.5, highlighting the necessity for additional enhancements. The model performed poorly for questions with no answer in the context, with a cosine similarity of only 0.17, suggesting that it is unsuitable for such scenarios. Our findings suggest that the proposed model is adequate for grading short answers, demonstrating consistency, and reducing bias by generating reliable reference answers. While the model's performance for long contexts and answers is acceptable, there is still room to improve grading accuracy across all question types.

References

[1] R. Qasrawi and A. Beniabdelrahman, “The Higher And Lower-Order Thinking Skills (HOTS and LOTS) In Unlock English Textbooks (1st And 2nd Editions) Based On Bloom’S Taxonomy: An Analysis Study,” *Int. Online J. Educ. Teach. (IOJET*, vol. 7, no. 3, 2020.

[2] W. H. Gomaa and A. A. Fahmy, “Ans2vec: A Scoring System for Short Answers,” in *Advances in Intelligent Systems and Computing*, 2020. doi: 10.1007/978-3-030-14118-9_59.

[3] A. Magooda, M. A. Zahran, M. Rashwan, H. Raafat, and M. B.

Fayek, "Vector based techniques for short answer grading," in *Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016*, 2016.

[4] V. Ramnarain-Seetohul, V. Bassoo, and Y. Rosunally, "Similarity measures in automated essay scoring systems: A ten-year review," *Educ. Inf. Technol.*, vol. 27, no. 4, 2022, doi: 10.1007/s10639-021-10838-z.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019. doi: 10.18653/v1/N19-1423.

[6] F. J. Muis and A. Purwarianti, "Sequence-to-Sequence Learning for Indonesian Automatic Question Generator," in *2020 7th International Conference on Advanced Informatics: Concepts, Theory and Applications, ICAICTA 2020*, 2020. doi: 10.1109/ICAICTA49861.2020.9429032.

[7] C. P. Carrino, M. R. Costa-Jussà, and J. A. R. Fonollosa, "Automatic Spanish translation of the SQuAD dataset for multilingual question answering," in *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 2020.

[8] N. Abadani, J. Mozafari, A. Fatemi, M. Nematbakhsh, and A. Kazemi, "ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0," *Int. J. Web Res.*, vol. 4, no. 1, 2021.

[9] C. van Toledo, M. Schraagen, F. van Dijk, M. Brinkhuis, and M. Spruit, "Exploring the Utility of Dutch Question Answering Datasets for Human Resource Contact Centres," *Inf.*, vol. 13, no. 11, 2022, doi: 10.3390/info13110513.

[10] A. Das and D. Saha, "Deep learning based Bengali question answering system using semantic textual similarity," *Multimed. Tools Appl.*, vol. 81, no. 1, pp. 589–613, 2022, doi: 10.1007/s11042-021-11228-w.

[11] E. B. Page, "Project Essay Grade: PEG," in *Automated Essay Scoring: A Cross-Disciplinary Perspective*, 2003. doi: 10.4324/9781410606860-12.

[12] E. B. Page, "Computer grading of student prose, using modern concepts and software," *J. Exp. Educ.*, 1994, doi: 10.1080/00220973.1994.9943835.

[13] M. A. Hearst, "The debate on automated essay grading," *IEEE Intell. Syst. their Appl.*, 2002, doi: 10.1109/5254.889104.

[14] T. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, vol. 25, no. 2–3, pp. 259–284, 1998, doi: 10.1080/01638539809545028.

[15] M. Zhang, S. Hao, Y. Xu, D. Ke, and H. Peng, "Automated essay scoring using incremental latent semantic analysis," *J. Softw.*, 2014, doi: 10.4304/jsw.9.2.429-436.

[16] R. Setiadi Citawan, V. Christanti Mawardi, and B. Mulyawan, "Automatic Essay Scoring in E-learning System Using LSA Method with N-Gram Feature for Bahasa Indonesia," in *MATEC Web of Conferences*, 2018. doi: 10.1051/matecconf/201816401037.

[17] S. A. Savittri, A. Amalia, and M. A. Budiman, "A relevant document search system model using word2vec approaches," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1898/1/012008.

[18] A. Amalia, D. Gunawan, Y. Fithri, and I. Aulia, "Automated Bahasa Indonesia essay evaluation with latent semantic analysis," *J. Phys. Conf. Ser.*, vol. 1235, p. 012100, 2019, doi: 10.1088/1742-6596/1235/1/012100.

[19] M. A. Hussein, H. Hassan, and M. Nassef, "Automated language essay scoring systems: A literature review," *PeerJ Comput. Sci.*, 2019, doi: 10.7717/peerj-cs.208.

[20] M. Cozma, A. M. Butnaru, and R. T. Ionescu, "Automated essay scoring with string kernels and word embeddings," in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018. doi: 10.18653/v1/p18-2080.

[21] M. Yamamoto, N. Umemura, and H. Kawano, "Automated essay scoring system based on rubric," *Stud. Comput. Intell.*, 2018, doi: 10.1007/978-3-319-64051-8_11.

[22] V. S. Kumar and D. Boulanger, "Automated Essay Scoring and the Deep Learning Black Box: How Are Rubric Scores Determined?," *Int. J. Artif. Intell. Educ.*, 2021, doi: 10.1007/s40593-020-00211-5.

[23] S. Xue, J. Zhang, J. Zhou, and F. Ren, "Robust Automated Essay Scoring by Using Attentive Capsule," in *Proceedings of 2022 8th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2022*, 2022. doi: 10.1109/CCIS57298.2022.10016365.

[24] S. Ge and X. Chen, "The application of deep learning in automated essay evaluation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020. doi: 10.1007/978-3-030-38778-5_34.

[25] Y. Sharma and S. Gupta, "Deep Learning Approaches for Question Answering System," in *Procedia Computer Science*, 2018. doi: 10.1016/j.procs.2018.05.090.

[26] A. H. Mohammed and A. H. Ali, "Survey of BERT (Bidirectional Encoder Representation Transformer) types," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1963/1/012173.

[27] E. R. Djoko, ; Rikel, M. ; Mansor, and R. Slater, "BERT for Question Answering on BioASQ," *SMU Data Sci. Rev.*, vol. 3, no. 3, 2020.

[28] C. Sung, T. Ma, T. I. Dhamecha, V. Reddy, S. Saha, and R. Arora, "Pre-training BERT on domain resources for short answer grading," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019. doi: 10.18653/v1/d19-1628.

[29] S. K. Gaddipati, D. Nair, and P. G. Plöger, "Comparative Evaluation of Pretrained Transfer Learning Models on Automatic Short Answer Grading," Sep. 2020, Accessed: Dec. 10, 2023. [Online]. Available: http://arxiv.org/abs/2009.01303

[30] S. Minaee and Z. Liu, "Automatic question-answering using a deep similarity neural network," in *2017 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2017 - Proceedings*, 2018. doi: 10.1109/GlobalSIP.2017.8309095.

[31] Y. Indrihapsari *et al.*, "A Comparison of OpenNMT Sequence Model for Indonesian Automatic Question Generation," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 8, no. 1, 2023, doi: 10.21831/elinvo.v8i1.56491.

[32] M. Fuadi and A. D. Wibawa, "Automatic Question Generation from Indonesian Texts Using Text-to-Text Transformers," in *Proceedings - IEIT 2022: 2022 International Conference on Electrical and Information Technology*, 2022. doi: 10.1109/IEIT56384.2022.9967858.

[33] M. H. Haidir and A. Purwarianti, "Short Answer Grading Using Contextual Word Embedding and Linear Regression," *J. Linguist. Komputasional*, vol. 3, no. 2, 2020.

[34] M. C. Wijaya, "Automatic Short Answer Grading System in Indonesian Language Using BERT Machine Learning," *Rev. d'Intelligence Artif.*, vol. 35, no. 6, pp. 503–509, 2021, doi: 10.18280/ria.350609.

[35] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[36] E. Mayfield and A. W. Black, "Should you fine-tune BERT for automated Essay scoring?," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.bea-1.15.

[37] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuad: 100,000+ questions for machine comprehension of text," in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016. doi: 10.18653/v1/d16-1264.

[38] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018. doi: 10.18653/v1/p18-2124.

[39] Z. A. Guven and M. O. Unalir, "Natural language based analysis of SQuAD: An analytical approach for BERT," *Expert Syst. Appl.*, vol. 195, 2022, doi: 10.1016/j.eswa.2022.116592.

[40] Y. Wu *et al.*, "Google's Neural Machine Translation System," *ArXiv e-prints*, 2016.

[41] F. Rashel, A. Luthfi, A. Dinakaramani, and R. Manurung, "Building an Indonesian Rule-Based Part-of-Speech Tagger.pdf," pp. 70–73, 2014.

[42] R. I. Pre-trained, "Named Entity Recognition in Electronic Medical Attention," *IAENG Int. J. Comput. Sci.*, vol. 51, no. 4, pp. 401–408, 2024.

[43] A. Luthfi, B. Distiawan, and R. Manurung, "Building an Indonesian named entity recognizer using Wikipedia and DBPedia," *Proc. Int. Conf. Asian Lang. Process. 2014, IALP 2014*, pp. 19–22, 2014, doi: 10.1109/IALP.2014.6973520.

[44] S. Faza *et al.*, "AUTOMATIC GENERATION OF MULTIPLE-CHOICE QUESTIONS USING TEMPLATE-BASED SEMANTIC WEB IN INDONESIAN LANGUAGE," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 2, 2024.

[45] S. M. Jain, "Hugging Face," in *Introduction to Transformers for NLP*, 2022. doi: 10.1007/978-1-4842-8844-3_4.

[46] A. Vaswani *et al.*, "[Transformer] Attention is all you need," *Adv.*

Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, 2017.

[47] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, “IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP,” in *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference,* 2020. doi: 10.18653/v1/2020.coling-main.66.

[48] T. Agrawal, “Optuna and AutoML,” in *Hyperparameter Optimization in Machine Learning*, 2021. doi: 10.1007/978-1-4842-6579-6_5.

[49] H. Brenner and U. Kliebsch, “Dependence of weighted kappa coefficients on the number of categories,” *Epidemiology*, vol. 7, no. 2, 1996, doi: 10.1097/00001648-199603000-00016.

[50] N. A. Kurdhi and A. Saxena, “Evaluating Quadratic Weighted Kappa as the Standard Performance Metric for Automated Essay Scoring,” *Int. Educ. Data Min. Soc.*, 2023.

[51] C. N. Tulu, O. Ozkaya, and U. Orhan, “Automatic Short Answer Grading with SemSpace Sense Vectors and MaLSTM,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3054346.

[52] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701.

[53] T. Wolf, L. Debut, V. Sanh, J. Chaumond, and ..., “HuggingFace’s Transformers: State-of-the-art Natural Language Processing,” *arXiv*, vol. 1910.03771, 2019.

[54] D. M. Williamson, X. Xi, and F. J. Breyer, “A Framework for Evaluation and Use of Automated Scoring,” *Educ. Meas. Issues Pract.*, vol. 31, no. 1, 2012, doi: 10.1111/j.1745-3992.2011.00223.x.