**International Journal of Informatics and Communication Technology (IJ-ICT)**
Vol. 14, No. 2, August 2025, pp. 478~487
ISSN: 2252-8776, DOI: 10.11591/ijict.v14i2.pp478-487 

# IndoBART optimization for question answer generation system with longformer attention

**Peter Andrew, Abba Suganda Girsang**
Department of Computer Science, Binus Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

**Article Info**

***Article history:***

Received Apr 30, 2024
Revised Dec 13, 2024
Accepted Jan 19, 2025

***Keywords:***

Fine-tuning
Natural language generation
Natural language processing
Question answer generation
Transformer

**ABSTRACT**

The Incorporation of Question Answering system holds immense potential for addressing Indonesia's educational disparities between the abundance of high school students and the limited number of teachers in Indonesia. These studies aim to enhance the Question Answering System model tailored for the Indonesian language dataset through enhancements to the Indonesian IndoBART model. Improvement was done by incorporating Longformer's sliding windows attention mechanism into the IndoBART model, it would increase model proficiency in managing extended sequence tasks such as question answering. The dataset used in this research was TyDiQA multilingual dataset and translated the SQuADv2 dataset. The evaluation indicates that the Longformer-IndoBART model outperforms its predecessor on the TyDiQA dataset, showcasing an average 26% enhancement across F1, Exact Match, BLEU, and ROUGE metrics. Nevertheless, it experienced a minor setback on the SQuAD v2 dataset, leading to an average decrease of 0.6% across all metrics.



***Corresponding Author:***

Peter Andrew
Department of Computer Science, Binus Graduate Program-Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
Email: peter.andrew@binus.ac.id

## 1. INTRODUCTION

There is an imbalance in the teacher and student ratio in Indonesia. Based on Indonesia Central Statistic, in 2021/2022, the number of high school students in Indonesia reached 10,063,926. While the available number of teachers was only 700,742. One teacher has to oversee around 14 students. With the advancement of technology, it could assist teachers in the learning process and ensure the effectiveness of the teacher effort [1].

In education, reading is an important activity. Reading enables students to understand the text or reading material and is a cognitive process for extracting information. Reading strategies, interpreting words, and reading techniques are also crucial for effective learning [2], [3]. Due to the importance of reading, teaching strategies have been developed to teach and evaluate reading [4]. One of the developed strategies is question generation (QG). However, the process of creating questions is time-consuming [5]. From this issue, the research on QG, also known as automatic question generation (AQG) [6], [7], has been developed as a branch of natural language generation (NLG).

QG is the process of automating the creation of questions based on textual input in various forms such as short answer, open-ended questions, multiple-choice, and fill-in-the-blank [8]. The aim of QG is to create natural questions to test the knowledge acquired from reading. Question Answer Generation was first introduced in 1999 at the text retrieval conference (TREC) as a test for system capabilities in finding short

and concise texts to answer given questions. The testing yielded satisfactory results as the system could extract specific texts that were predicted to best answer the given questions. Question answer research has rapidly developed and combined several different but related research fields, including information extraction (IE), natural language processing (NLP), and information retrieval (IR) [9].

The AQG system typically consists of three conceptual processes [8]. Which is target selection, the stage of identifying important sentences and keywords. Question representation construction, determining the type and syntactic form of questions based on the sentences and keywords contained in the text. Question realization, the final stage of question creation.

Question answering systems can be divided into two types [9]. Closed domain, the system can only handle questions in specific fields (sports, politics, and health). It can also be interpreted as a condition where there are limitations on question types, such as descriptive questions. This type can be facilitated by using NLP systems with the exploitation of specific scientific fields (ontologies). Open domain, the system can handle general questions that are not limited to specific fields of knowledge. Such systems usually require a large amount of data to obtain accurate answers.

There are several approaches to question-answering systems, including [9], frequently asked questions and answers (FAQs), The easiest approach involves collecting a dataset of question-answer pairs stored in the system. When a question is given, the system searches for the answer from the stored dataset. Information Retrieval, this is the most commonly used approach, where the main concept is to search for accurate and precise answers from a collection of documents. The general steps used in this approach include pre-processing, question analysis, document retrieval, and answer extraction. Machine learning, this approach is similar to Information Retrieval, but with the addition of classification algorithms to classify question types.

Several developments in AQG systems have been made. Here are some related AQG research studies such as, the development of the stanford question answering dataset (SQuAD) by [10]. The dataset was collected from Wikipedia articles, with crowdsourced question-answer pairs. The human-validated test results achieved an exact match (EM) of 77% and an F1 Score of 86.8%. The development of a document retrieval method using a recurrent graph-based model by [11]. It was developed to solve the problem of entity-centric questions caused by information compression. The method achieved an improved F1 score of 2.9 and EM of 3.5 using the SQuAD dataset. The development of a document retrieval method utilizing Generative Models by [12]. It aimed to test the capability of Generative Models in searching for text documents containing evidence. The results showed an F1 score of 63.2 and EM of 56.7 on the SQuAD dataset, as well as an F1 score of 56.7 and EM of 80.1 on the TriviaQA dataset. The development of Decomposed Pre-Train Transformer by [13], with the goal of speeding up the process by 4.3x compared to previous transformer models, with only a 1% decrease in accuracy. In [14] introduced IndoNLG, a benchmark for NLG in three commonly used Indonesian languages: Bahasa Indonesia, Sunda, and Jawa. IndoNLG includes six evaluation tasks: machine translation (MT), QG, summarization, Chit-chat, and more. The dataset Indo4B-Plus [15] is used to pre-train the IndoBART and IndoGPT models, which achieve competitive results with 1/5th of the parameters compared to multi-lingual models like mBART.

Among the developed AQG systems, only a few have focused on the Indonesian language. Therefore, this article aims improving the accuracy of Indonesian-based AQG systems model by modifying the attention layer of the model using longformer – sliding windows attention.

## 2. RELATED WORKS

BLEU (Bilingual Evaluation Understudy) [16] compares MT to human translations, aiming to closely mirror professional human translations. BLEU scores individual translated segments against reference translations, then averages scores across the corpus to gauge overall translation quality. It prioritizes precision over factors like intelligibility and grammatical correctness by computing the presence of n-grams in the candidate translation also found in reference translations. BLEU offers variants such as BLEU-1 and BLEU-4, where n varies from 1 to a specified N (e.g., 1 or 4), and aggregates scores using a geometric mean.

Recall-oriented understudy for gisting evaluation (ROUGE) [17] is a robust metric used in NLP to evaluate automatic summarization and MT systems. It compares generated summaries or translations with human references, providing valuable insights. Notably case-insensitive. ROUGE-L variants, computes its score based on the longest common subsequence (LCS) between reference and candidate texts, offering flexibility without requiring a predefined n-gram size.

Several question-answering models have been developed to enhance accuracy and efficiency in handling general domain tasks, [18] proposed utilizing multiple frameworks to learn from representations, which can be divided into several processes. The model is based on the pre-trained model (XLNet) and subsequently fine-tuned using various datasets. The experimental results from different frameworks yielded an average EM score of 56.59 and an F1 score of 68.98 (from test datasets: BioProcess,

ComplexWebQuestions, MCTest, QAMR, QAST, TREC), indicating an improvement in accuracy compared to the BERT-large baseline model.

Decomposed transformer model [13] enhances time and memory efficiency, in order to reduce the complexity of the transformer function, it is decomposed based on input segmentation. Evaluation using the SQuAD v1.1, RACE, BoolQ, MNLI, and QQP datasets demonstrate significant improvement in speed and memory efficiency up to 4x increases. BART pre-trained model [19], which combines Bi-directional and Auto-Regressive Transformer models is a denoising autoencoder with a sequence-to-sequence model. The model achieves the highest performance on the SQuAD dataset using text infilling denoising method with precision of 90.8 F1 score.

Chung *et al*. [20] proposes a procedure for creating multilingual vocabularies by combining separately trained vocabularies from multiple language cluster derivations. This approach aims to balance the exchange between cross-lingual subwords and language-specific vocabulary. The evaluation results of this method using the TyDiQA, XNLI, and WikiAnn NER datasets show varying performance scores depending on the language. However, on average, the method achieves an F1 score of 70.1, which is 2.1 points higher than the initial joint method's F1 score of 68.0.

Optimization on transformer self attention layer to accommodate longer sequences length by changing the dot product mechanism on the self attention layer with sliding windows technique proposed in Longformer [21]. The self attention mechanism has the complexity of O(n2), while the proposed longformer model had the linear complexity of O(n). The evaluation shows an improvement by 3 points on Wikihop, 1 point on TriviaQA and HotpotQA compared to the RoBERTa-base model.

IndoNLG [14], a benchmark for NLG in three commonly used Indonesian languages: Bahasa Indonesia, Sunda, and Jawa. IndoNLG includes six evaluation tasks: MT, QG, Summarization, Chit-chat, and more. The dataset Indo4B-Plus is used to pre-train the IndoBART and IndoGPT models, which achieve competitive results with 1/5th of the parameters compared to multi-lingual models like mBART.

AQG system development for the Indonesian language [22] using the SQuAD v2 dataset, which has been translated using the Google Translate API, along with additional TyDiQA dev data. The proposed method utilizes a sequence-to-sequence approach and implements BiGRU, BiLSTM, and Transformer models.

## 3. METHOD

This study aims to enhance the state-of-the-art IndoBART model for the Question Answer Generation task. This enhancement involves the substitution of IndoBART self-attention mechanism with a sliding window attention inspired by Longformer [21]. The research is organized into three sequential stages: Data Preprocessing, Model Modification, and Training and Evaluation. The holistic workflow is visually depicted in Figure 1.

Collecting data: Based on Figure 1, SQuAD V2 [10] and TyDiQA [23] public datatset would be collected from public website such as kaggle or hugginface. SQuAD v2 consist of 130.000 train and 11.900 validation question-answer data, TyDiQA on the other hand consist of 151.000 train and 18.700 validation question-answer data. SQuAD v2 dataset is an english dataset, therefore translation to Indonesia language is completly needed. In contrast TyDiQA is a multillingual dataset that consist of 11 languages (telugu, arabic, swahili, japanese, finnish, Indonesia, russian, thai, korean, bengali, english). The percentage of Indonesia language in tydiqa is 9%, approximately 13.000 train and 1683 evaluation. Hence, for TyDiQA dataset translation step are ommited and only Indonesian dataset being used.

Translating english data: SQuAD v2 dataset would be translated using google translate API. Since the dataset question, answer and context paragraph attribute are translated seperately, some of the question-answer contexts were missing. That being the case, answer context matching mechanism [24] are applied as seen in Figure 2. Fuzzy string matching are being used to find the translated answer on the translated context paragraph, unmatch answer would be replace with empty string and -1 value are append to the translated_answer_start_position key, match answer location would be appended to the translated_answer_start_position key.

Data preprocessing: In data preprocessing, both TyDiQA and translated SQuAD v2 without an answer or having -1 value on answer start position would be removed. Special token would be added to add more context before the training, <BOS> token was used to mark beginning of sentence, <EOS> token was used to mark end of sentence, <SEP> token was used to seperate each context on the sentence. The overall structure of the input text was <BOS> [paragraph context] <SEP> [question] <EOS>. Input would be truncated if the length exceeds max_length hyperparameter, on the otherhand padding would be added with to keep consistent length of the input using max_length strategy to the rest of the dataset. Proccessed dataset would be tokenized on word unit using IndoNLG tokenizer (indoBART tokenizer). Figure 3 illustrate the data processing of model input.

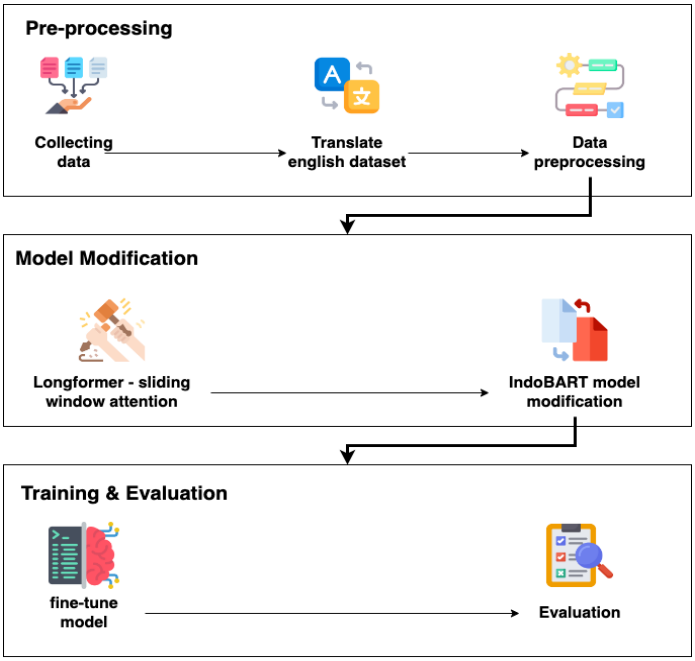



Figure 1. Research flow diagram

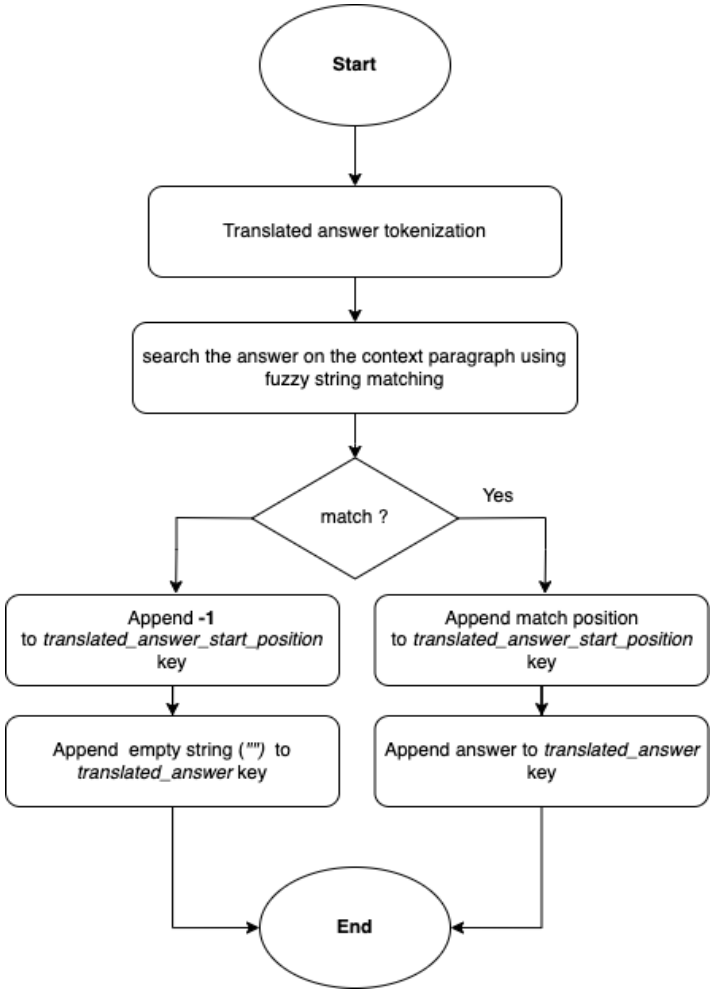



Figure 2. Answer-context matching flow

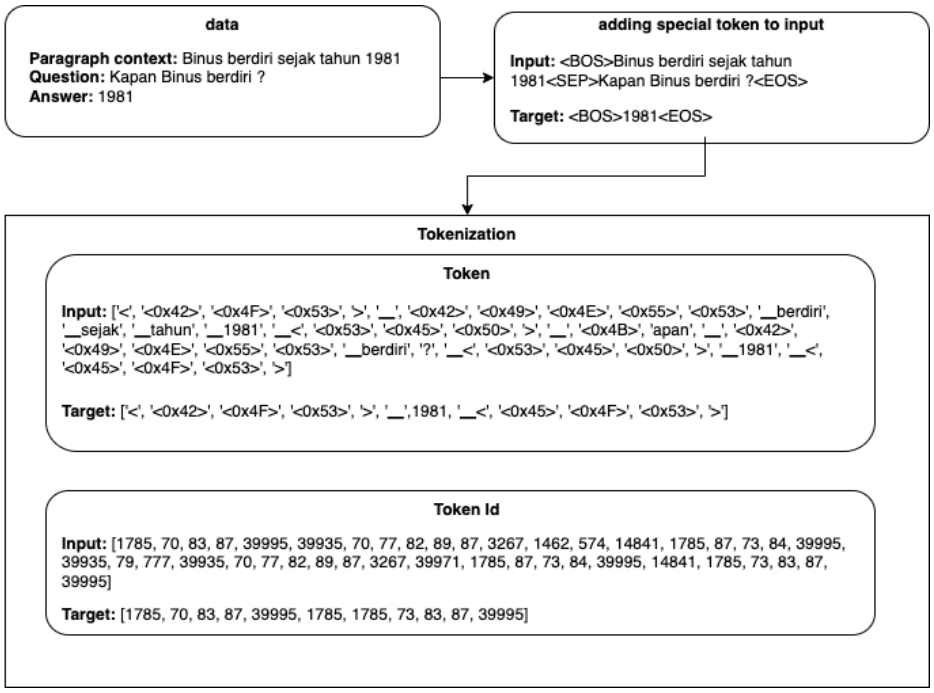
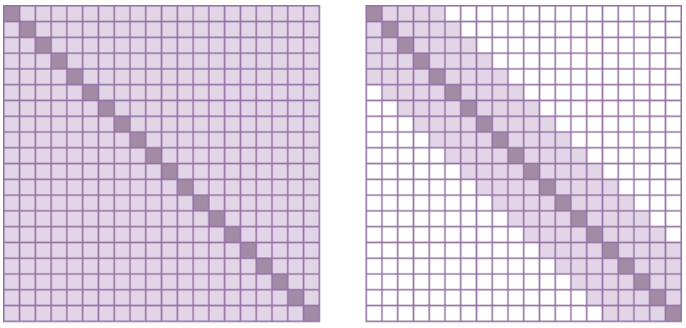



Figure 3. Data processing example

**Longformer – sliding window attention:** On this step, sliding window attention modification were implemented based on Longformer [1]. Conventional transformer self-attention [25] employs the matrix dot product computation, depicted in (1), resulting in $O(n^2)$ computation complexity for each layer. In (1) attention score.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Conversely, the sliding windows attention pattern computes only neighboring tokens with the fixed-size window (w), resulting in O(n * w) or O(n) computation complexity. This fixed-size window scale down from higher layer to lower layer. For this particular research, the window size was halved (1/2 w) with each descent in layer. In the higher layers, a larger window size was apply to capture high-level information, while the lower layers focused on capturing local contextual information. Additionally, The reduction in window size serves to maintain the equilibrium between efficiency and performance, smaller window sizes incur lower computational cost because they contain fewer nonzero values, thus enhancing efficiency, while larger window sizes possess greater representation capacity and often result in performance improvements. The computation comparison between self-attention and sliding window attention could be seen in Figure 4, Figure 4(a) illustrate operation on computing attention using self-attention whereas, Figure 4(b) shown operation on computing attention using sliding window attention.



(a) (b)

Figure 4. Attention mechanism comparison illustration (a) self-attention – $n^2$ and (b) sleding window attention – n

IndoBART model modification: IndoBART consist of six layers each for the encoder and decoder, with each layer possessing its own self-attention module. Therefore, the Longformer sliding window attention module is employed to replace each of these layers. To preserve the pre-trained weights of the IndoBART self-attention module, the weight values are copied to each Longformer attention module before the replacement occurs. Both the base model and the Longformer-modified model are fine-tuned using the processed dataset, questions and context paragraphs serve as input, while answers are used as output.

Fine-tune model: Both the base and modified models are being fine-tuned separately for each dataset. During the fine-tuning phase, the training dataset is split with an 80:20 ratio between training and testing data. Data has already been tokenized in a prior process utilizing IndoNLGTokenizer, making it ready to be directly used as input for the model. The model generates tokenized output, which undergoes a detokenization process using IndoNLG to convert it into string format. To ensure fair benchmarking, both models are fine-tuned using the same hyperparameters as presented in Table 1. The Longformer-modified model has the autoregressive feature disabled by default, while IndoBART possesses the autoregressive feature. Therefore, this research will conduct Longformer-modified training twice, once with and once without the autoregressive feature, to ensure thorough study.

Table 1. Hyperparameter

| Hyperparameter | Value |
|---|---|
| Tokenizer embedding size | 1024 |
| Learning rate | 2e-5 |
| Dropout rate | 0.1 |
| Training batch | 3 |
| Evaluation batch | 4 |
| Training epoch | 2 |

**Evaluation:** Each model will undergo separate training and evaluation using the respective datasets. Model accuracy will be assessed using F1, Exact Match, BLEU, and ROUGE metrics. Tracking training time will be implemented to benchmark the optimization of attention layer complexity. Further evaluation details are available in section 4.3.

## 4. RESULT AND DISCUSSION

### 4.1. Data preprocessing

The SQuAD v2 dataset are translated from english language to Indonesia language using google translate API, Example of translated SQuAD v2 and TyDiQA could be seen in Table 2. Then translated to make sure translated answer and question have a macthing context on the context paragraph, Fuzzy string matching are performed, to compare the answer with context paragraph. When the answer doesn't match any string in the context paragraph the data row are deleted. After performing fuzzy string matching, there is a lot of data point that didn't have macthing answer on the context paragraph, resulting in dataset size reduction from 130.000 train data to 82.000 data and 11.900 validation data to 10.900 data.

For the TyDiQA dataset, original dataset already provided some data in Indonesia language, hence dataset that are used in this research only took the indonesian language portion of the dataset. The number of data in Indonesia language are 13.500 train data and 1.800 validation data. Answer validation are performed to make sure every data point have an answer on the context paragraph. Upon validating, the number of data that didn't had an answer are a lot, resulting in quite huge data reduction to 5.700 train data and 550 validation data. Then the data are being formatted to spesific format to make a uniform learning pipeline with SQuAD Dataset. Example of TyDiQA dataset that are being used in this research could be seen in Table 3.

Table 2. Translated SQuAD v2 example

| Title | Context | Question | Indonesian_answer |
|---|---|---|---|
| Beyonce | *Beyoncé Giselle Knowles-Carter (/ biːˈjɒnseɪ / bee-YON-say) (lahir 4 September 1981) adalah penyanyi, penulis lagu, produser dan aktris rekaman Amerika...* | *Kapan Beyonce mulai menjadi populer?* | *{ "answer_start": 291, "text": "pada akhir 1990-an" }* |
| Kota New York | *New York — sering disebut New York City atau Kota New York untuk membedakannya dari Negara Bagian New York...* | *Kota apa di Amerika Serikat yang memiliki populasi tertinggi?* | *{ "answer_start": 0, "text": "New York" }* |
| Frédéric Chopin | *Frédéric François Chopin (22 Februari atau 1 Maret 1810 - 17 Oktober 1849), kelahiran Fryderyk Franciszek Chopin ...* | *Bagaimana kewarganegaraan Frédéric?* | *{ "answer_start": 201, "text": "Polandia dan komposer Prancis" }* |

Table 3. TyDiQA Indonesia example

| Title | Context | Question | Indonesian answer |
|---|---|---|---|
| Fernando de Magelhaens | *Sayangnya Raja Portugis John, terbunuh pada tahun 1495 dan Pangeran Manuel, yang lebih berminat akan harta sebaliknya daripada penjelajahan, naik takhta...* | *Kapan perdagangan melalui armada mulai dilakukan oleh bangsa Eropa ?* | *{ "answer_start": 453, "text": "1505" }* |
| Bandar Pasir Mandoge, Asahan | *Bandar Pasir Mandoge adalah sebuah kecamatan di Kabupaten Asahan, Sumatera Utara, Indonesia.* | *Dimana letak daerah "Pardembanan"?* | *{ "answer_start": 49, "text": "Kabupaten Asahan, Sumatera Utara, Indonesia" }* |
| Gurun Sahara | *Sahara terletak di utara Afrika dan berusia 2,5 juta tahun. Padang pasir ini membentang dari Samudra Atlantik ke Laut Merah...* | *Berapakah Luas Gurun Sahara ?* | *{ "answer_start": 437, "text": "9.000.000km2" }* |

### 4.2. Model fine-tuning

The input and output samples for each model are presented in Table 4 for the SQuAD v2 dataset and Table 5 for the TyDiQA dataset. In both instances, the input comprises two primary attributes: the "context paragraph" and the "question," which are delineated by a <sep> token to facilitate separation. Conversely, the output pertains to the "answer" attribute.

Table 4. SQuAD v2 input, target and output comparison example

| Model | Example |
|---|---|
| Input | <BOS> *Segera setelah Normandia mulai memasuki Italia, mereka memasuki Kekaisaran Bizantium dan kemudian Armenia, bertempur melawan Pecheneg, Bulgaria, dan terutama Turki Seljuk. Tentara bayaran Norman pertama kali didorong untuk datang ke selatan oleh orang-orang Lombard untuk bertindak melawan Bizantium, tetapi mereka segera berperang dalam dinas Bizantium di Sisilia. Mereka menonjol di samping kontingen Varangian dan Lombardia dalam kampanye Sisilia George Maniaces pada 1038–40. Ada perdebatan apakah orang Normandia dalam dinas Yunani sebenarnya berasal dari Norman Italia, dan sekarang tampaknya hanya sedikit yang datang dari sana. Juga tidak diketahui berapa banyak "kaum Frank", sebagaimana orang-orang Bizantium menyebutnya, adalah orang-orang Normandia dan bukan orang Prancis lainnya.<SEP>Siapa yang berperang melawan Normandia di Italia?*<EOS> |
| Target | *Pechenegs, para Bulgaria, dan terutama orang-orang Turki Seljuk* |
| IndoBART-base output | pecheneg, bulgaria |
| Longformer-IndoBART output | pecheneg |
| Longformer-IndoBART – autoregressive output | pecheneg, bulgaria |

Table 5. TyDiQA input, target, and output comparison example

| Model | Example |
|---|---|
| Input | <BOS>Hiragana (Kana: ひらがな; Kanji: 平仮名*) adalah suatu cara penulisan bahasa Jepang dan mewakili sebutan sukukata. Pada masa silam, ia juga dikenali sebagai* onna de (女手) *atau 'tulisan wanita' karena biasa digunakan oleh kaum wanita. Kaum lelaki pada masa itu menulis menggunakan tulisan Kanji dan Katakana. Hiragana mulai digunakan secara luas pada abad ke-10 Masehi.*<SEP>*apakah yang dimaksud dengan Hiragana*?<EOS> |
| Target | *cara penulisan bahasa Jepang dan mewakili sebutan sukukata* |
| IndoBART-base output | *penulisan abad ke-10 masehi* |
| Longformer-IndoBART output | *cara penulisan bahasa jepang* |
| Longformer-IndoBART – autoregressive output | *cara penulisan bahasa jepang* |

### 4.3. Evaluation

The evaluation phase were benchmark using the F1-score, Exact Match, BLEU, and ROUGE evaluation metric for each model and dataset. Evaluation result could be seen in Table 6 for SQuAD dataset and Table 7 for TyDiQA dataset. IndoBART model base on mBART model which by default were using auto-regressive feature of BART, On the other hand the longformer attention had the option to disabled the auto-regressive of the model, therefore autoregressive would be on of the variation of the modified model.

Table 6. SQuAD V2 evaluation result

| Model name | Exact match | F1 | BLEU | ROUGE-L | ROUGE-1 | ROUGE-2 |
|---|---|---|---|---|---|---|
| IndoBART-base | **35.06** | **54.04** | **0.054** | **0.529** | **0.53** | **0.282** |
| **Longformer-IndoBART** | 34.81 | 53.48 | 0.0531 | 0.524 | 0.525 | 0.277 |
| **Longformer-IndoBART - autoregressive** | 34.91 | 53.68 | 0.0536 | 0.526 | 0.527 | 0.28 |

Table 7. TyDiQA evaluation result

| Model name | Exact match | F1 | BLEU | ROUGE-L | ROUGE-1 | ROUGE-2 |
|---|---|---|---|---|---|---|
| IndoBART-base | 28.49 | 39.94 | 0.061 | 0.39 | 0.39 | 0.27 |
| **Longformer-IndoBART** | 34.69 | 48.92 | 0.081 | 0.48 | 0.48 | 0.31 |
| **Longformer-IndoBART - autoregressive** | **34.86** | **50.81** | **0.1** | **0.50** | **0.50** | **0.34** |

Based on Table 6, base-indoBART model perform slightly better than longformer-indoBART model across all evaluation metric on translated SQuAD v2 dataset, most significant difference of 0.4 seen in F1-score. Whereas, longformer-IndoBART achieved outstanding result on TyDiQA dataset (Table 7) in contrast to base-indoBART. Longformer-IndoBART outmatch base-indoBART model on every evaluation metric, longformer-IndoBART most notable advancement shown in F1-score, it outperform base-indoBART by 25%. Longformer-indoBART model with autoregressive appear to perform better then without autoregressive on both dataset, It shown autoregressive feature generally perform better on Question Answering task.

TyDiQA generally had longer input paragraph sequence and output answer sequence, in those condition longformer-indoBART model could perform better than base-indoBART model. Moreover, some context might lost in translation for SQuAD v2 dataset. In this cases, base-indoBART full self-attention presumably have better context understanding then sliding-window attention on longformer-indoBART model, as it check all of the sequences compared to checking on the sliding window range.

The graph in Figures 5 and 6 illustrates a comparison of training times across all models for each dataset. Figure 6 indicate the training time for the SQuAD dataset, where the IndoBART-base model trained approximately 1.5 times faster than the modified Longformer-IndoBART model, the autoregressive feature in Longformer incurred a slightly longer training time than its non-autoregressive counterpart. Turning to Figure 5, which illustrates the training time for the TyDiQA dataset, reveal the IndoBART-base model trained nearly twice as quickly as the Longformer-IndoBART modification, with the Longformer's autoregressive feature taking a fraction of a second longer than its non-autoregressive counterpart.

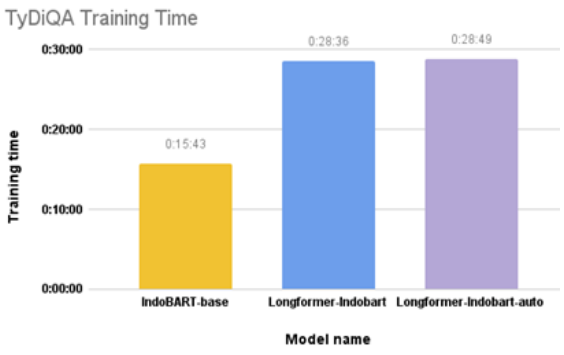



Figure 5. TyDiQA training time comparison

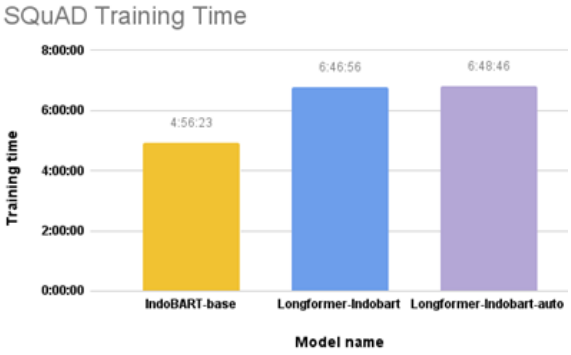



Figure 6. SQuAD training time comparison

In theory, the Longformer modification boasts superior computational complexity of $O(n)$ compared to the IndoBART-base self-attention, which stands at $O(n^2)$. However, empirical findings from training time evaluations reveal that the Longformer modification actually took longer than the base IndoBART. This discrepancy can be attributed to the optimized implementation of matrix dot product computations in the IndoBART-base, which proved to be faster than the unoptimized sliding windows technique utilized in the Longformer modification. Thus, while the Longformer modification indeed involves a lower number of operations, its unoptimized implementation resulted in longer training times compared to IndoBART self-attention.

## 5. CONCLUSION

To conclude, in accordance of the evidence presented in this research, utilization of Longformer attention modification to improve Indonesia-based AQG system models is notably beneficial, especially when handling longer sequence context input and output. Throughout this research, the Longformer - IndoBART model showcased remarkable precision across all evaluative metrics on the TyDiQA dataset in contrast to its predecessor, the IndoBART-base model. While, the Longformer-IndoBART model

demonstrated impressive performance, it experienced a slight setback when evaluated on the translated SQuAD v2 dataset compared to the IndoBART model, because of some context went missing in translation.

For future studies, it is advisable to conduct training with QG as the target instead of answer generation. Furthermore, employing an unsupervised learning approach to generate both questions and answers from a contextual paragraph could streamline the manual process of teacher-led question-answer generation. The datasets employed in this study were sourced from either multilingual or English datasets that required translation. This could pose challenges as some context may be lost in translation or the dataset size may be limited. Therefore, the creation of an Indonesian language question-answering dataset is essential to develop more robust models and benchmarks.

## FUNDING INFORMATION

Authors state no funding involved.

## AUTHOR CONTRIBUTIONS STATEMENT

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Peter Andrew | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | |
| Abba Suganda Girsang | ✓ | | | | | ✓ | | | ✓ | ✓ | ✓ | ✓ | | ✓ |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest

## DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article

## REFERENCES

[1] N. T. Le, T. Kojiri, and N. Pinkwart, "Automatic question generation for educational applications - the state of art," in *Advances in Intelligent Systems and Computing*, vol. 282, Springer International Publishing, 2014, pp. 325–338.
[2] R. Smith, P. Snow, T. Serry, and L. Hammond, "The Role of background knowledge in reading comprehension: a critical review," *Reading Psychology*, vol. 42, no. 3, pp. 214–240, Feb. 2021, doi: 10.1080/02702711.2021.1888348.
[3] E. B. Moje, P. P. Afflerbach, P. Enciso, and N. K. Lesaux, *Handbook of reading research, volume V*, vol. V. Routledge, 2020.
[4] M. Leasa, A. Abednego, and J. R. Batlolona, "Problem-based learning (PBL) with reading questioning and answering (RQA) of preservice elementary school teachers," *International Journal of Learning, Teaching and Educational Research*, vol. 22, no. 6, pp. 245–261, Jun. 2023, doi: 10.26803/ijlter.22.6.14.
[5] V. Ramadayanti and Jamiluddin, "Developing students' reading comprehension through question generation strategy," *E-Journal of English Language Teaching Society*, vol. 8, 2020.
[6] X. Du, J. Shao, and C. Cardie, "Learning to Ask: Neural Question Generation for Reading Comprehension," CoRR, vol. abs/1705.00106, 2017, [Online]. Available: http://arxiv.org/abs/1705.00106
[7] D. Suhartono, M. R. N. Majiid, and R. Fredyan, "Towards automatic question generation using pre-trained model in academic field for Bahasa Indonesia," *Education and Information Technologies*, vol. 29, no. 16, pp. 21295–21330, Apr. 2024, doi: 10.1007/s10639-024-12717-9.
[8] V. Harrison and M. Walker, "Neural generation of diverse questions using answer focus, contextual and linguistic features," in *INLG 2018 - 11th International Natural Language Generation Conference, Proceedings of the Conference*, 2018, pp. 296–306, doi: 10.18653/v1/w18-6536.
[9 T. Sultana and S. Badugu, "A review on different question answering system approaches," in *Learning and Analytics in Intelligent Systems*, vol. 4, Springer International Publishing, 2020, pp. 579–586.
[10] 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, pp. 2383–2392, doi: 10.18653/v1/d16-1264.
[11] A. Asa, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over wikipedia graph for question answering," *arXiv.*, 2019.
[12] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, 2021, pp. 874–880, doi: 10.18653/v1/2021.eacl-main.74.

[13] Q. Cao, H. Trivedi, A. Balasubramanian, and N. Balasubramanian, "DeFormer: Decomposing pre-trained transformers for faster question answering," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4487–4497, doi: 10.18653/v1/2020.acl-main.411.
[14] S. Cahyawijaya *et al.*, "IndoNLG: benchmark and resources for evaluating indonesian natural language generation," in *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2021, pp. 8875–8898, doi: 10.18653/v1/2021.emnlp-main.699.
[15] B. Wilie *et al.*, "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 843–857, 2020.
[16] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002, vol. 2002-July, pp. 311–318, doi: 10.3115/1073083.1073135.
[17] C.-Y. Lin, "ROUGE: a package for automatic evaluation of summaries," *Text Summarization Branches Out, Barcelona, Spain: Association for Computational Linguistics*, pp. 74–81, 2019.
[18] D. Su *et al.*, "Generalizing question answering system with pre-trained language model fine-tuning," in *MRQA@EMNLP 2019 - Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019, pp. 203–211, doi: 10.18653/v1/d19-5827.
[19] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.
[20] H. W. Chung, D. Garrette, K. C. Tan, and J. Riesa, "Improving multilingual models with language-clustered vocabularies," in *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2020, pp. 4536–4546, doi: 10.18653/v1/2020.emnlp-main.367.
[21] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: the long-document transformer," *arXiv preprint arXiv*.
[22] 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*, Sep. 2020, pp. 1–6, doi: 10.1109/ICAICTA49861.2020.9429032.
[23] J. H. Clark *et al.*, "Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 454–470, Dec. 2020, doi: 10.1162/tacl_a_00317.
[24] K. Vincentio and D. Suhartono, "Automatic question generation using RNN-based and pre-trained transformer-based models in low resource indonesian language," *Informatica (Slovenia)*, vol. 46, no. 7, pp. 103–118, Nov. 2022, doi: 10.31449/inf.v46i7.4236.
[25] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

## BIOGRAPHIES OF AUTHORS

**Peter Andrew** is currently freelance developer specializing in web, mobile, desktop, and cross-platform development since 2024, He holds a bachelor's degree in informatics from Bina Nusantara University 2021. He has an internship at the Apple Developer Academy in 2019 as an iOS developer, followed by a transition to Flash Coffee in 2021, where he demonstrated proficiency in mobile, web, and backend development, alongside adeptness in build automation and data reporting. his area of interest is machine learning and data science. He can be contacted at email: peter.andrew@binus.ac.id.

**Abba Suganda Girsang** is currently a lecturer at Master in Computer Science, Bina Nusantara University, Jakarta, Indonesia Since 2015. He got Ph.D. degree in 2015 at the Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. He graduated bachelor from the Department of Electrical Engineering, Gadjah Mada University (UGM), Yogyakarta, Indonesia, in 2000. He then continued his master's degree in the Department of Computer Science at the same university in 2006–2008. He was a staff consultant programmer in Bethesda Hospital, Yogyakarta, in 2001 and also worked as a web developer in 2002–2003. He then joined the faculty of the Department of Informatics Engineering in Janabadra University as a lecturer in 2003-2015. His research interests include swarm, intelligence, combinatorial optimization, and decision support system. He can be contacted at email: agirsang@binus.edu.