Education and Information Technologies (2024) 29:21295–21330
https://doi.org/10.1007/s10639-024-12717-9

Towards automatic question generation using pre-trained model in academic field for Bahasa Indonesia

Derwin Suhartono[1] · **Muhammad Rizki Nur Majiid**[1] · **Renaldy Fredyan**[1]

Received: 1 May 2023 / Accepted: 13 April 2024 / Published online: 30 April 2024

Abstract

Exam evaluations are essential to assessing students' knowledge and progress in a subject or course. To meet learning objectives and assess student performance, questions must be themed. Automatic Question Generation (AQG) is our novel approach to this problem. A comprehensive process for autonomously generating Bahasa Indonesia text questions is shown. This paper suggests using a decoder to generate text from deep learning models' tokens. The suggested technique pre-processes Vectorized Corpus, Token IDs, and Features Tensor. The tensors are embedded to increase each token, and attention is masked to separate padding tokens from context-containing tokens. An encoder processes the encoded tokens and attention masks to create a contextual understanding memory that the decoder uses to generate text. Our work uses the Sequence-to-Sequence Learning architecture of BiGRU, BiLSTM, Transformer, BERT, BART, and GPT. Implementing these models optimizes computational resources while extensively exploring the research issue. The model uses context sentences as input and question sentences as output, incorporating linguistic elements like response placement, POS tags, answer masking, and named entities (NE) to improve comprehension and linguistic ability. Our approach includes two innovative models: IndoBERTFormer, which combines a BERT encoder with a Transformer decoder, and IndoBARTFormer, which decodes vectors like BERT. IndoTransGPT uses the Transformer as an encoder to improve understanding, extending the GPT model's adaptability.

Keywords Educational assessment · Question generation · Pre-trained language model · Long short-term memory · Gated recurrent unit · Transformer

Extended author information available on the last page of the article

1 Introduction

The final exam is still suitable for evaluating learning by having students respond to diverse kinds of questions to gauge their degree of knowledge. Adaptive learning is a concept employed widely at many levels of education, with elementary schools being one such place (Vie et al., 2017). Not only does the availability of an adaptive learning process change the learning model, but it also changes how learning is evaluated (Al-Chalabi et al., 2021). Modifying questions based on students' ability distinguishes the adaptive learning model's evaluation procedure from standard learning models. Therefore, it is tough for teachers to produce questions of varied types and degrees of complexity. Understanding the subject matter, choosing the subject matter for the questions, and producing questions with varying degrees of difficulty all require a thorough process. Previous studies have shown that not all Indonesian teachers can create questions of varying degrees of difficulty, particularly those requiring Higher-Order Thinking Skills (HOTS) (Blegur et al., 2023). To address these issues, educational processes have been radically altered by incorporating information and communication technologies (ICT), providing novel answers. For this section, we have reviewed the literature on information and communication technology (ICT) in education at length on multiple occasions.

Incorporating new educational technology has become essential in educational institutions due to the rapid evolution of the educational landscape. This paradigm shift comprises a range of creative approaches, including e-learning and m-learning, which have fundamentally changed the way teaching and learning work. Educational institutions have shown considerable interest in the implementation of mobile learning applications. The study conducted by Almaiah & Al Mulhem provides a comprehensive analysis of the factors influencing the decision to use mobile learning applications in universities. The study compares universities that have implemented mobile learning with those that have not (Almaiah & Al Mulhem, 2019). This comparative approach illuminates the crucial factors influencing mobile learning application adoption, offering significant insights for educational institutions. Moreover, adopting e-learning technologies has demonstrated its ability to bring about significant changes. The same authors introduce a theoretical structure for identifying the key elements that contribute to the effectiveness of implementing e-learning systems, employing the Delphi technique (Almaiah & Al Mulhem, 2019). This framework provides a beneficial guide for educational institutions implementing e-learning systems. It helps them grasp the critical variables that contribute to the success of these systems and offers a roadmap for their journey toward transformation.

To assess the influence of these technologies, it is necessary to consider the viewpoints of the leading players, namely the students. The study conducted by Almaiah et al. examined students' viewpoints about mobile learning services. By considering students' perspectives, we can obtain valuable insights about the effects and usability of mobile learning from the standpoint of the individuals who use it. A thorough comprehension of mobile learning covers its utilization

and creation (Almaiah & Jalil, 2014). In other publications, they undertake a thorough analysis examining how numerous elements influence the growth of mobile learning applications at divergent phases of usage. Their study contributes to comprehending the aspects that impact the growth and progression of mobile learning applications, which is a crucial element of incorporating technology in education (Almaiah et al., 2020).

Although these technologies hold immense potential, they also present significant obstacles. (Amin Almaiah & Jalil, 2016) Present a persuasive case study that examines the significant issues and actions involved in implementing mobile learning systems. This study offers a practical perspective on educational institutions' difficulties while adopting mobile learning technology. The COVID-19 epidemic led to an extraordinary increase in the use of mobile learning devices in educational institutions. Studies conducted by Almaiah et al. examine the factors that influence the utilization of mobile learning applications during this moment of transformation. These studies provide insights into the ability to adapt and the difficulties encountered by universities in Jordan. As we continue to examine the connection between education and technology, we will utilize this research to further our comprehension of the difficulties and critical elements for successfully using e-learning and m-learning technologies in educational institutions. These ideas will inform our research and add to the continuing discussion on the influential capabilities of educational technology (Almaiah et al., 2021, 2022).

Furthermore, questions used as evaluation materials can accurately determine student understanding results, and the quality of the questions must be maintained. Well-crafted questions inspire students to use their imaginations and relate knowledge while responding to them. The primary factor in developing HOTS-type questions is the relationships between pieces of information (Abosalem, 2015). The instructor must be able to relate knowledge manually before processing it into questions of the HOTS style; however, not all teachers are adept at doing so, making it challenging to maintain the consistency of questions (Kusuma et al., 2022). That is, even while the material utilized to create the questions is the same, the questions generated by each person would differ. It is vital to automate the question-generation process due to this diversity (Papasalouros & Chatzigiannakou, 2018).

Several techniques have been proposed for an Automatic Question Generation (AQG) system based on NLP in the NLG branch (Mazidi & Tarau, 2016). All proposed models have covered rule-based till attention-based ones (Alsubait et al., 2016). According to published research, AQG automatically generates pertinent questions from various input forms, including text, a structured database, or a knowledge base. In various fields, including Massive Open Online Courses, AQG can be applied naturally to solve question-generation processes for educational purposes (Kurdi et al., 2020). The two primary types of questions are objective questions and subjective questions. The objective question invites students to select the best response out of multiple options or to offer a word or brief phrase to complete a statement or to respond to a question. Multiple-choice, matching, true–false, and fill-in-the-blank questions are examples of objective questions, while subjective questions call for an explanation as a response (Annamoradnejad et al., 2020).

This study focuses on the Automated Question Generation utilizing language models based on Transformers to aid or support educators in developing Reading Comprehension (RC) tests (Rogers et al., 2023). Since teachers may spend less time on mundane tasks and more time with their students, creating engaging experiences for in-person classroom interactions is topical and vital. As a result, this would help lessen students' worry about the online learning environment: their isolation from peers and teachers (Palvia et al., 2018).

The research question of this study is how to use advanced natural language processing models, namely IndoBERT and IndoGPT, to make an Automated Question Generation (AQG) system that works well for Reading Comprehension (RC) tests. The problem statement is about how hard it is for teachers to produce good questions that regularly assess students' knowledge and higher-order thinking skills. Using AQG systems based on NLP, we aim to speed up the process of producing questions and make educational tests more useful. By answering this research question, we will critically analyze AQG implementation challenges in multilingual educational contexts, emphasizing Bahasa Indonesia. We investigate and evaluate AQG models to understand the obstacles, opportunities, and best practices for using AQG in educational evaluations.

As shown in Fig. 1, this study explores the complex world of Automatic Question Generation (AQG) from both Question Generation and Question Answering perspectives. We use well-known pre-trained IndoBERT (Koto et al., 2020) and IndoGPT (Cahyawijaya et al., 2021), utilizing advanced deep-learning techniques. These language models have undergone careful refinement to oversee tasks involving question generation (QG) and question answering (QA). Our research uses the strength of these language models to develop an end-to-end pipeline for AQG. In addition, we suggest a novel architectural strategy based on IndoBERT and

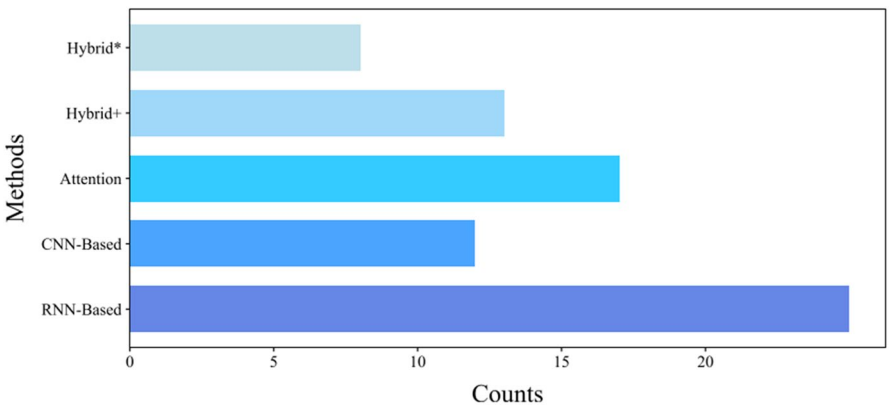

Fig. 1 Statistics on techniques of deep learning. Hybrid + refers to the strategies that combine several deep learning architectures. Hybrid* approaches combine deep learning structures with non-DL techniques (Hao et al., 2022). Hence, the X axis might depict numerous research endeavors that have attempted to tackle this problem by integrating non-deep learning techniques into the architecture. However, deep learning approaches still need to consistently provide the highest performance for specific tasks still possess some advantages. The pairings can synergistically leverage individual capabilities

IndoGPT, using them as assessment benchmarks. Our model receives a multidimensional input, including the sentence, question, answer, and supporting information. The duty of predicting the question text is then given to it, which aligns with the methodology used in the SQuAD-ID and TyDiQA-ID datasets. We use rigorous metrics, such as BLEU (Papineni et al., 2002) and ROGUE (Lin, 2004), to measure the effectiveness of AQG and give a thorough evaluation of the questions created.

Implementing an entire Question Generation (QG) system, including creating the training code and improving the QG model, is one of this study's main contributions. (1) Creating a thorough processing pipeline that enables end-to-end question generation by utilizing optimized IndoBERT and IndoGPT language models is one of these contributions. (2) The introduction of a novel method for Automatic Question Generation (AQG) based on the Text-To-Text paradigm. Although this method was implicitly discussed before, it will be fully described in the following sections for a clearer understanding. (3) Evidence that our strategy, based on the previously mentioned language models and uses the Text-To-Text paradigm, not only effectively generates questions but also produces findings that stand up to comparison with other approaches.

The remainder of this paper is divided into five sections. Part 1 gives the information that this research is essential to enhance teaching–learning in the education field. Moreover, Part 2 provides an overview of the current state of the art for the most recent neural language models, their applicability to tasks like QG and QA, and an examination of the current implementations. The implementation of the end-to-end pipeline is described in Section 3 of the paper's materials and methods section, which describes the architecture based on the IndoBERT, IndoGPT, and mT5 models. Section 4 offers the results and comments for analysis and a review of our approach. Part 5 summarizes the contributions and provides recommendations for more studies in its conclusion.

2 Recent studies

2.1 Natural language understanding

Natural language comprehension in contextual understanding is often quite challenging due to the complexity of the subject and the relative dearth of conversational training data (Hunter et al., 2019). Consequently, limited and domain-specific training data usually limit the ability of NLU components in context systems to generalize (Shigehalli, 2020). Developing an NLU component that performs well in low-data situations is challenging when just a few samples are provided for each system-specific goal. Language models trained on vast unlabeled datasets, such as Bidirectional Encoder Representations from Transformers (BERT), have recently attained modern performance on natural language processing tasks (Devlin et al., 2019). These large-scale, pre-trained language models provide contextualized word embeddings and include transferrable linguistic properties such as syntactic chunks and parts of speech (Liu et al., 2019a, 2019b).

Owing to the advantages of employing pre-trained language models, the BERT model is the most often studied NLU technique. BERT has been explicitly modified for two NLU applications: aim classification and text similarity. In the suggested intent classifier model, a pre-trained BERT model is created on top of Bidirectional Long Short-Term Memory (BiLSTM), which is then tweaked. Nevertheless, optimizing BERT on a restricted data set may result in overfitting, which degrades performance. The proposed intent classifier combines phrase representation from the text similarity model with BERT + BiLSTM to improve performance on a small dataset and in few-shot circumstances. They use the given text corpus and similarity model to detect which text the user references while posting a remark. Using a considerable supervised semantic textual similarity (STS) (Cer et al., 2017) benchmark dataset, the text similarity model outperforms the BERT model for the textual similarity problem.

2.2 Natural language generation

Creating meaningful phrases, sentences, and paragraphs from an internal representation is known as natural language generation (NLG), which differs from NLU. NLG is a part of NLP and comprises four stages: goal identification, goal planning using situation analysis, communication resource evaluation, and plan realization as a text (Khurana et al., 2023). In recent years, hidden Markov models, state machines, and rule-based implementations have given way to neural language models for natural language processing (NLP) and natural language understanding (NLU) (Jurafsky, 2000).

The NLP research community accepted the use of the Transformer after it was developed because of its effectiveness (Radford et al., 2019). This has made it possible to train larger models more effectively, which has led to the development of new performance measurements and, in some cases, human performance outperformance, such as in RC benchmarks like SQuAD (Raffel et al., 2020). The Transformer is quickly replacing other training and language model development architectures. Some of these language models are also accessible in pre-trained form and exhibit remarkable performance in tasks like QA are T5, GPT-2, BERT, ALBERT (A Lite BERT) (Raffel et al., 2020), and RoBERTa (Robustly optimized BERT Pretraining Approach) (Raffel et al., 2020).

2.3 Automatic question generation

In recent years, there have been two distinct AQG sequence-to-sequence architectural techniques. First, the majority of study has used Recurrent Neural Networks (RNN) with their bidirectional (Sundermeyer et al., 2014), either Long Short-Term Memory (LSTM) (Yao & Guan, 2019) or Gated Recurrent Unit (GRU) (Cho et al., 2014) equivalent of backpropagation, while the second method uses Transformer (Vaswani et al., 2017). The SQuAD dataset was used in most of the investigations to build all these models, as in previous works. Aside from that, the evaluation system used in these investigations needed more consistency. In their first stage, (Serban

et al., 2016) used a knowledge graph to create a model input representation. The dataset is used to build the knowledge graph, in which the subject of the question is associated with the topic of the fact, and the link between the fact and the answer to the question is associated with the object. They used the 100,000 question–answer pairs in the Simple Question dataset (Bordes et al., 2015). They created the sequence-to-sequence architecture to adhere to the structure of translation tasks. The encoder is known as a sub-encoder, and it can recover three inputs, subject, relationship, and object atoms, which are then combined to form fact embedding. Before training the decoder, the encoder was trained independently.

Using Bahdanau et al.'s model (Bahdanau et al., 2015) and Luong et al.'s improvements (Luong et al., 2015, Du et al., 2017) introduced a new architecture that uses beam search and a simplified copy mechanism to deal with words that are not in the vocabulary (Garneau et al., 2019). These words are replaced by words from the source sentence carefully instead of UNK (unknown) tokens. Harrison and Walker (Harrison & Walker, 2018) also developed their own AQG system based on LSTM for both the encoder and decoder. The use of linguistic features, such as Named Entity (NE), Coreference Resolution, Binary Cased Word Indicator (case), and Part-of-Speech Tag (POS), distinguishes this model from others. See's copy mechanism is also utilized by this system (See et al., 2017). On the other hand, Kumar et al. (Kumar et al., 2018) developed a novel framework. They used something called a "generator-evaluator," which is a brand-new method. By developing a novel loss function, this reinforcement learning system significantly improves backward propagation and weight updating. The generator adds language and response placement capabilities to Bahdanau's sequence-to-sequence architecture.

The AQG approach developed by Liu et al. (B. Liu et al., 2019a, 2019b) utilized a novel clue indicator that facilitates question generation. A Graph Convolutional Network is trained to produce clue indication (GCN). This hint is the input for the Bi-LSTM-LSTM sequence-to-sequence architecture and its additional characteristics. Moreover, (Dong et al., 2019) investigate that Transformer is one of four language model (LM) types in the AQG system. This model accepts the token location, sentence segment, and word-embedded token. The four LMs with three distinct types of mechanism attention are built using the output from the parameter-sharing Transformer that is used to analyze those inputs.

Numerous studies on AQG based on NLP using the English language have been conducted with various languages to address various issues, including those in education, medicine, and other fields (Zhang et al., 2017)(Akyön et al., 2022). However, Indonesia needs more studies due to its limited resources and inability to create a large dataset.

A study that was done in Indonesian created a language model that uses a sequence-to-sequence methodology and is trained on the TyDiQA dataset (Clark et al., 2020) and the SQuAD v2 dataset (Rajpurkar et al., 2016), which were translated into Indonesian using the Google Translate API v2 to the model with the Transformer architecture and Recurrent Neural Network (RNN) such as the Bi-LSTM, Bi-GRU, and Transformer from scratch. Moreover, by comparing innovative multilingual models like mBART and mT5 with monolingual models like IndoBART and IndoGPT, Vincentio and Suhartono (Vincentio & Suhartono, 2022)

analyzed the TyDiQA and SQuAD datasets. As a result, fine-tuned IndoBART outperformed the RNN-based and other pre-trained models.

Furthermore, IndoGPT was built using the GPT-2 architecture in more depth, but it was adapted to the Bahasa Indonesia dataset. Besides, GPT -2 (Radford et al., 2019) is the successor to the first GPT model (Radford et al., 2018), which has ten times the number of parameters and was trained on an even larger pre-training dataset. The architectural style and training program of the GPT model is still present in the GPT-2 model. In addition to the well-known applications in Question Answering, this paper explores a Text-to-Text approach for generating question text using the IndoBERT, IndoGPT, and mT5 language models. It suggests an end-to-end pipeline for Question Generation based on these models.

3 Materials and methods

3.1 Datasets

Our research utilizes three highly valuable datasets: SQuAD (Stanford Question Answering Dataset), TyDiQA (Typologically Diverse Question Answering), and IDK-MRC (Indonesian Machine Reading Comprehension). Each of these datasets is crucial in improving our research and assessment procedures, providing diverse contributions to the question generating and responding field. These datasets were selected for our study based on their distinct characteristics and the valuable insights they offer. Together, they enhance the extent and variety of our research, presenting various difficulties and qualities. Our diverse range of experiences allows us to have a broad understanding of how effective question-generation strategies are in different linguistic and cultural situations. By integrating these datasets into our research, we aim to enhance the significance and practicality of our findings. Every data set has unique complexity and nuances, enabling us to investigate the diverse environment of question generation thoroughly.

3.1.1 SQuAD

Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) is a reading comprehension dataset comprising crowd-sourced inquiries on various Wikipedia pages on several topics. The original SQuAD dataset comprises more than 100,000 question–answer pairs and 23,215 paragraphs from 536 articles. Alternatively, a question may not have an answer at all. Furthermore, the newest update, SQuAD 2.0, combines questions from SQuAD 1.1 with more than 50,000 unanswerable questions that crowd workers have crafted to seem as though they are answered (Rajpurkar et al., 2018). To do effectively on SQuAD2.0, systems must respond to questions when they can, recognize when the paragraph supports no response, and choose not to respond.

Relating to this dataset, we found that Muis and Purwarianti (Muis & Purwarianti, 2020) investigated AQG using an Indonesia dataset based on SQuAD v2.0 translated from English to Indonesian using the Google Translate API v2 for about

536 articles and 161.550 question–answer pairs. Due to the size of the data, no translation changes were made, and they utilized the translation outcome precisely as it was. Instead, most of the dataset post-processing activity focuses on changing the answer's value and answer location. The answer to each question is a span of tokens from the relevant reading passage. Furthermore, we drop unanswerable questions to ensure the data is noise-free. The quantity of the splitting is shown in Table 1.

3.1.2 TyDiQA

Typologically Diverse Question Answering (TyDiQA) (Clark et al., 2020) is a multilingual question–answer dataset containing 200,000 pairs of human-annotated questions and answers in 11 typologically varied languages. Each question is matched with a Wikipedia page, just as a Natural Question (NQ). The model must make two predictions: an index of the passage that answers the query (Passage Selection Task); and the shortest possible span that fully answers the question (Minimal Answer Span Task). Like NQ, TyDiQA offers a blind test set and keeps a scoreboard with the same evaluation metrics.

3.1.3 IDK-MRC

Indonesian Machine Reading Comprehension (IDK-MRC) is the next level of MRC dataset that can be compared to other available MRC datasets, such as Translating SQuAD and TyDiQA, Putri and Oh's (Putri & Oh, 2022) findings highlight the usefulness of this dataset in enhancing the MRC models' capacity to handle unanswerable questions. In addition, they declare that the automated dataset creation process may expedite data collecting and lower associated costs. Following the human inspection, machine-generated concerns about dataset noise and imbalanced question types are diminished.

Although it is meant to produce unanswerable questions for Indonesians, this dataset-gathering technique may also be used in other medium-to-low resource languages or other quality assurance tasks, such as adversarial question formulation. Although the same pipeline (automatic creation, validation, and human generation)

Table 1 The research utilized datasets, which included the dataset titles, the number of paragraphs in each dataset, and the count of Question–Answer pairings (QA-pairings). In addition, it separates the QA-Pairs into training and testing subsets. The training subset is used to train the model, while the testing subset is used to evaluate its performance. This table provides a thorough overview of the dataset's composition and organization, making it easier to understand the data's extent and its importance to the study

Datasets	Paragraph		QA-Pairs	
	Train	Test	Train	Test
SQuAD-ID	19,035	1,204	120,054	10,885
IDK-MRC	2,9927	732	7,440	1,892
TyDiQA-ID	4,561	1,141	4,561	1,141

may be used for other languages or QA activities, the process for automatically creating questions must still be modified.

3.2 Pre-processing pipeline

To achieve our research objectives, we have developed a clear pipeline that methodically refines raw data into a polished and experiment-ready dataset. This pipeline consists of several crucial steps, each advancing the approach. Getting question–answer pairings from multiple sources is the first stage. We use strict filtering standards to ensure our dataset is of the highest caliber. We specifically omit blank or unanswered questions because they would be counterproductive to our research goals. After the initial filtering phase, the data is tokenized using the Natural Language Toolkit (NLTK) module. A crucial first step in getting the text data ready for analysis is tokenization. It divides the material into more manageable chunks, usually words or phrases. The data then passes via embedding, a crucial change after tokenization. For this, we use a pre-trained FastText model with 300 dimensions. This model is a potent tool for transforming our text data into numerical representations because it was previously trained on the sizable Common Crawl Dataset in Indonesian. These embeddings are essential for later modeling and analysis. The high-level view of the research pipeline is shown in Fig. 2.

In our early investigation, we also concentrated on the issues associated with these programming languages, as the vocabulary of our encoder was created on a dataset pertaining exclusively to Python. We note that only the questions with the interrogative phrases "how, what, why, which, when, and is/are/was/were" were retained in the dataset. Moreover, questions with a single token and an undetermined intent, such as "dd," are dropped. In addition, we saw a similar pattern on the answer feature, such as "H" and "G," and we assumed that it was noise. However, 96% in SQuAD-ID, 52% in IDK-MRC, and 96% in TyDiQA-ID of the high-quality question–answer combinations in our filter fit this constraint.

Moreover, the NLTK (*Natural Language Toolkit*) tokenizer needs help to effectively segregate unique tokens in code snippets, which results in an extensive vocabulary and aggravates the out-of-vocabulary problem. The ASCII charset has three distinct categories of printable characters: punctuation marks, numerals (0 to 9), and letters (A/a to Z/z). To solve this issue, we chose a tokenizing algorithm from various

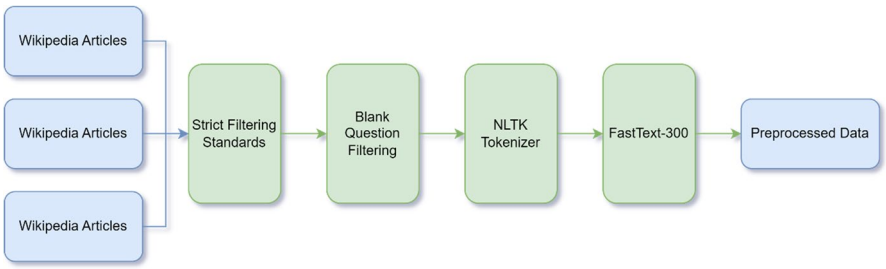

Fig. 2 High-level view of the research pipeline

pre-trained models, including FastText with Wikipedia, Tatoeba, and SETimes data, all in Bahasa, Indonesia. When tokenizing a string, we insert white spaces on each side of punctuation marks before dividing the string into tokens. In this approach, we get lengthier sequences in exchange for a representative vocabulary.

The pre-processing stages have been visually concluded in Fig. 3. The pre-processing flowchart outlines a methodical process for preparing input data for subsequent tasks, using distinct color codes for each component. The blue portions serve as contextual information, setting the background for further investigations. The green color emphasizes important entities in the text, aiding in recognizing and comprehending significant aspects. Orange represents Part-of-Speech (POS) tags, aiding in linguistic analysis and syntactic comprehension. Yellow parts indicate responses, critical areas for understanding and making inferences. Purple represents token IDs, which allow for the effective representation and handling of textual data. The red segments represent the combined tensor of context, entities, POS tags, and replies, enabling comprehensive processing and analysis of incoming data. This thorough pre-processing methodology, directed by the color-coded flowchart, guarantees vital data preparation, establishing a firm basis for future analytical pursuits.

Textual data requires preprocessing before being used in machine learning tasks. These steps can vary depending on the data's type and source. Entity Extraction and Representation identifies and categorizes specific information like names, dates, and types from the raw data. These extracted entities are then converted into numerical representations called NER tensors, making them easier for machine learning models to process. The text is then broken down into individual words or "tokens." This step may also involve removing frequently-used words with little meaning, like "the" or "and" called stop words or other unnecessary elements. Finally, each word

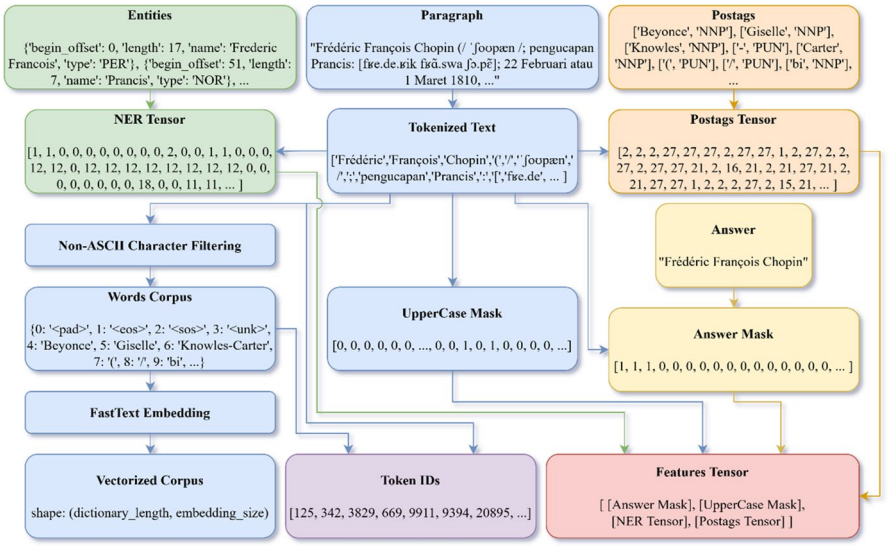

Fig. 3 Pre-processing flowchart, **Blue:** context, **Green:** entity, **Orange:** POS tags, **Yellow:** answers, **Purple:** token IDs, **Red:** concatenated tensor of context, entity, POS tags, and answers

is embedded into a numerical vector using FastText. This vector captures the word's semantic meaning, allowing the model to understand the relationships between words. These embedded words are then organized into a vectorized corpus, a structured format suitable for machine learning models.

When dealing with a raw paragraph, different preprocessing steps are applied. First, the paragraph is broken down into smaller units called tokens. These tokens can be individual words, punctuation marks, or other meaningful units. Specific features like punctuation, special characters, and uppercase letters are extracted while preserving their original order. Each token is assigned a part-of-speech tag, such as a noun or verb. These tags are then converted into a numerical representation called a postags tensor, which helps analyze the sentence structure. Next, specific answers are extracted from the postage tensor based on specific criteria or questions. An answer mask is created to identify these answers within the original text, highlighting their positions.

A comprehensive feature set is created by combining the answer mask, the uppercase mask (highlighting uppercase letters), the NER tensor, and the postags tensor. This combined feature set provides a rich representation of the text and feeds into various machine-learning models for tasks like classification or prediction.

Handling exceptionally long sequences still needs to be done in deep learning. So, the maximum length of the sequence that can be used will be limited so that later, the model does not slow down in learning. Figures 4 and 5 show the distribution of the sequence length ranges in the dataset; the sequence ranges in red bars will be trimmed according to the 99% quantile of the entire dataset (white bars). Table 2 displays the statistics of our processed dataset. Next, we modified the architecture by combining pre-trained models with Transformers to improve model understanding.

3.3 Masked language modeling

When considering language models, it is crucial to recognize the underlying constraints of specific models. Models such as BERT have several drawbacks when it comes to tasks like generating text and calculating the probabilities of sentences. Consequently, these models are frequently repurposed to serve as "authentic" language models, which are commonly known as Pretrained Language Models (PLM) (Salazar et al., 2020). They are mostly used to initialize encoder-decoder models in generative jobs. To provide more clarity, let us explore the Masked Language Modeling (MLM) concept in greater detail. MLM, a crucial element in models such as BERT, can be understood as a method to achieve stochastic *maximal pseudolikelihood estimation* (MPLE). When working with a training set $\mathbb{W}$,

it is essential to consider that the variables $\left\{\boldsymbol{w}_t\right\}_{t=1}^{|\boldsymbol{w}|}$ create a completely linked graph. This comes close to the traditional MLE, with MLM training asymptotically maximizing the goal:

$$\mathcal{J}_{PL}(\Theta;\mathbb{W}) = \frac{1}{|\mathbb{W}|}\sum_{W\in\mathbb{W}} PLL(W;\Theta)$$

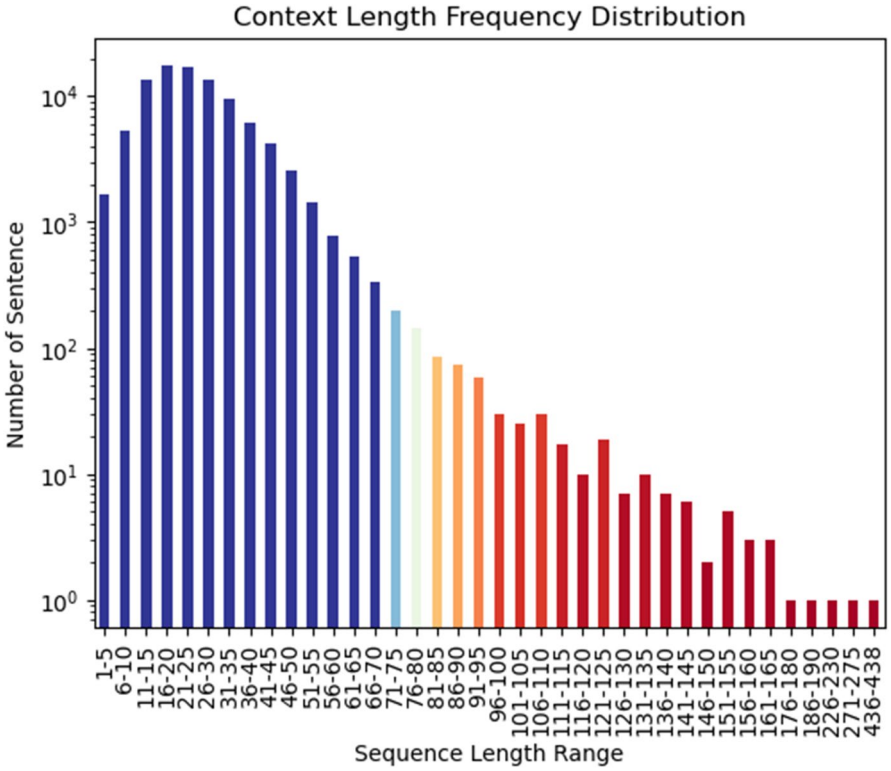

Fig. 4 Context length frequency distribution

By masking at position t, conditional distributions $\boldsymbol{w}_t|\boldsymbol{W}_{\setminus t}$ are described by an underlying joint distribution that MLMs learn. This made it possible to generate text using Gibbs sampling with BERT, which led to the suggestion (but not assessment) of a related number, the sum of logits, for sentence ranking. Earlier work on future conditional LMs in ASR was expanded with highly bidirectional self-attentive language models in more recent work. They used the [MASK] scoring approach to train shallow models from scratch. However, they did not connect their work to pseudolikelihood and fluency, which provide a framework to explain their results and observed behaviors.

Within the context of IndoBERTFormer, the Masked Language Model (MLM) will be employed in Section 4.1.4. Masked Language Modeling (MLM) is an important pre-training activity for the BERT model; hence its incorporation is necessary. Using MLM, IndoBERTFormer can improve its performance and language understanding by learning the fundamental language representations from contextual clues in masked tokens.

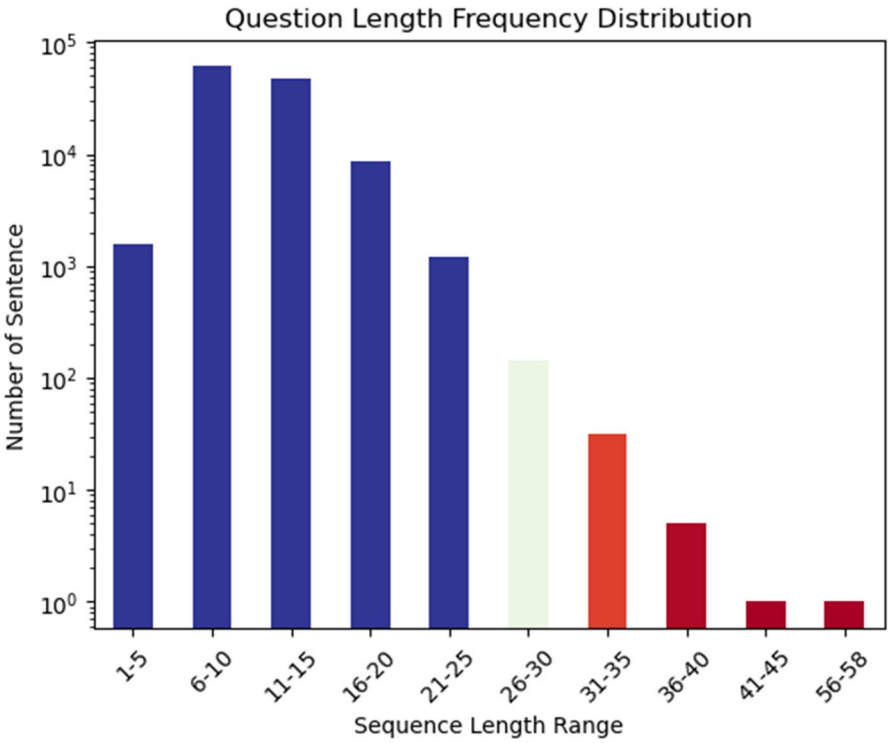

Fig. 5 Question length frequency distribution

Table 2 This table provides a thorough summary of datasets used in the research, following important data filtering and pre-processing steps to exclude extraneous data, unrelated answers, and unknown answers. Every dataset is comprehensive, providing its name or identity, the number of paragraphs kept, and the total count of Question–Answer pairs (QA-Pairs)

Datasets	Paragraph		QA-Pairs	
	Train	Test	Train	Test
SQuAD-ID	19,035	1,204	116,421	10,536
IDK-MRC	2,927	732	3,881	977
TyDiQA-ID	4,561	1,141	4,403	1,098

3.4 Paragraph and answer encoder

The prolonged response Network has a standard encoder that can separately encode the passage X^p and the longer answer X^s. We initially encode the joint word, answer position, NER, and POS embeddings at the paragraph level as $w^p = (w_1^p, \ldots, w_{n_p}^p)$, where n_p is the paragraph length $X^p, w_i^p \in \mathbb{R}^{d_w, d_a, d_n, d_p}$, and d_w, d_a, d_n, d_p is the dimensionality of the corresponding context, POS embeddings, NER, and answer position. We employ a bidirectional RNN and many pre-trained models to encode the paragraph, which takes

w^p as input and outputs the forward and backward hidden states P, to capture more context information. These are combined to represent a paragraph as $P = p_1, \ldots, p_{n_p}$. For time-step i of the text, this elaborate encoding procedure is determined as follows:

$$p_i = \left[\vec{p}_i; \overleftarrow{p}_i\right]$$

3.5 Paragraph and answer decoder

A paragraph decoder and an answer decoder are components of a question-answering system, a type of natural language processing system designed to answer questions posed in natural language. The paragraph decoder is responsible for processing a given passage of text (the "paragraph") and producing a vector representation of its meaning. This vector representation can then be used to compare the paragraph to a given question and identify the most relevant parts of the paragraph to include in the answer. Conversely, the answer decoder takes the information gathered from the paragraph decoder and produces an actual answer to the question.

This can be done using various methods, such as selecting the most relevant sentence or phrase from the paragraph, generating a new sentence that summarizes the information in the paragraph, or using a pre-defined database of answers. Overall, the paragraph decoder and answer decoder work together to enable a question-answering system to understand and respond to natural language questions, making it a powerful tool for information retrieval and natural language processing applications. The decoder's hidden state is initialized as follows:

$$h_0 = \tanh(\boldsymbol{W}_0 \overleftarrow{p}_1 + b)$$

where $\overleftarrow{p}_1$ is the latest hidden state of the backward paragraph encoder.

The last word is embedding w^y_{t-1} and the context vector $\widetilde{p}_{t-1}$ are input the architecture decoder uses at each decoding step t to generate the new hidden state h_t.

$$h_t = \boldsymbol{Architecture}(h_{t-1}, \left[w^y_{t-1}; \widetilde{p}_{t-1}\right]$$

By the concatenate attention mechanism (Luong et al., 2015), the context vector $\widetilde{p}_t$ for the current time-step t is calculated as follows:

$$e_{t,i} = v^{\mathrm{T}}$$

$$\alpha^d_{t,i} = \frac{\exp\left(e_{t,i}\right)}{\sum_{j=1}^{n_p} \exp\left(e_{t,j}\right)}$$

$$\widetilde{p}_t = \sum_{i=1}^{n_p} \alpha^d_{t,i} \widehat{p}_i$$

where $\alpha_{t,i}^{d}$ is the significance score matching the current decoder state h_t with each encoded paragraph representation $\widehat{p}_i$ and $e_{t,i}$ is the normalized attention weight on the encoded paragraph representation $\widehat{p}_i$ at current time-step.

The previous word embedding w_{t-1}^{y}, the current context vector $\widetilde{p}_t$, and the current decoder state h_t are used to generate the readout state r_t, which is then transmitted via a maxout hidden layer (Goodfellow et al., 2013) to construct the probability distribution of the next word over the decoder vocabulary.

$$r_t = \boldsymbol{W}_r w_{t-1}^{y} + \boldsymbol{U}_r \widetilde{p}_t + \boldsymbol{V}_r h_t$$

$$m_t = [\max\{r_{t,2j-1}, r_{t,2j}\}]_{j=1,\ldots,d}^{\mathsf{T}}$$

$$p_v(y_t|y_{<t}) = softmax(\boldsymbol{W}_m m_t)$$

3.6 Proposed training model

Many neural network models have been put out because of deep learning advancements to encode language for autonomous analysis and improved representation at the sentence level. Encoding textual data using deep learning models into the pre-trained model is another way to close the knowledge gap. Moreover, to produce some text, we need a decoder to generate text based on the model's token, as seen in Fig. 6.

Vectorized Corpus, Token IDs, and Features Tensor are the three tensors that underwent pre-processing. Each of them will be sent to the encoding layer. To enlarge the dimensions of each token, vectorized corpus, and token IDs must first pass through the embedding layer to purchase embedded tokens. These embedded tokens would be intended to be focused on tensor characteristics to create tensor input. Moreover, an attention mask separates a token padding token from the context-containing token. The encoder, which produces a memory that contains the understanding of the context's outcomes, may then be used to encode both the input tensor and the attention mask. The decoder is awaiting this memory and the target tensor's processing.

It should be noted that the target tensor at the beginning of the iteration only comprises matrices with zero values. This decoder will create an embedded token that corresponds to one ID token. This will replace zero values in the target tensor depending on the order. The decoding process will repeat itself after the end-of-sequence (EoS) token is acquired.

The prepared dataset served as a basis for developing our study's encoder and decoder components, which were then organized according to the Sequence-to-Sequence Learning architecture. Our investigation used several algorithms, including BiGRU, BiLSTM, Transformer, BERT, BART, and GPT. These models were implemented to ensure the research subject was thoroughly explored while maximizing the use of computer resources. The central design used context

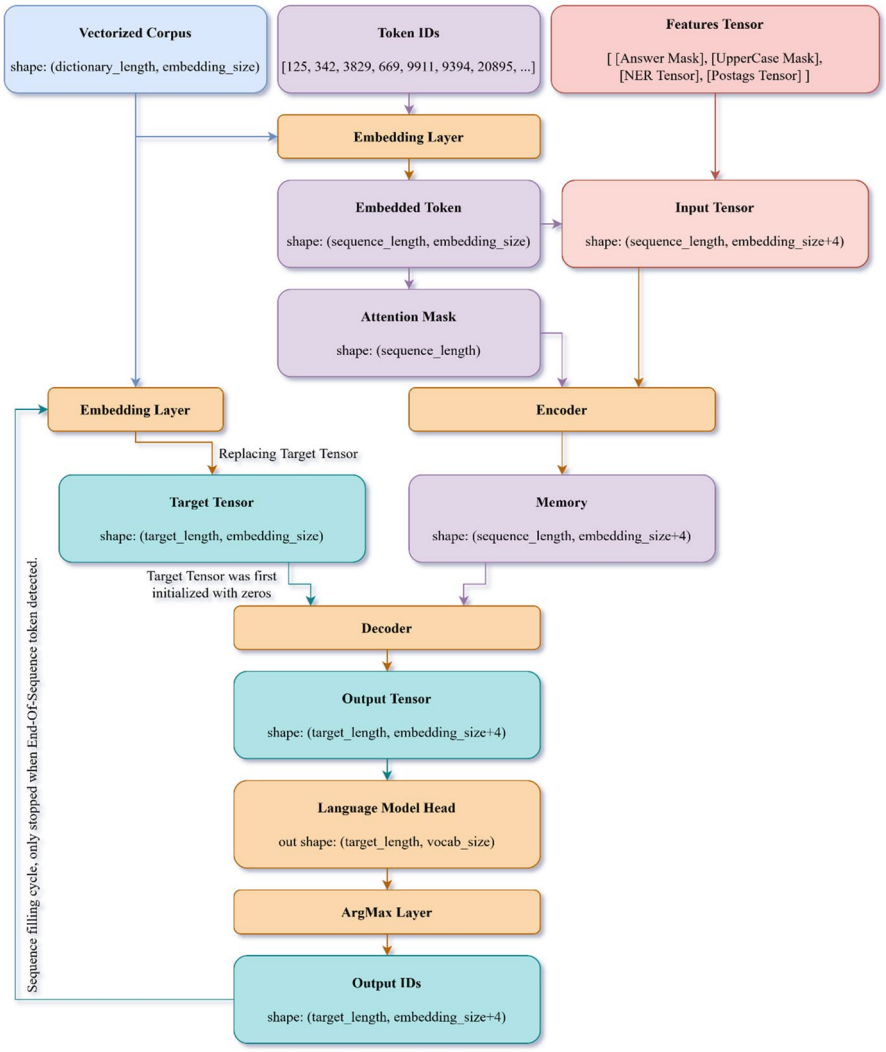

Fig. 6 Proposed architecture of indonesian automatic question generation

sentences from paragraphs as the input and question sentences as the goal output. We included many linguistic features, such as answer location (ans), part of speech (POS) tags, answer masking, and named entities (NE), to enhance the model's comprehension and linguistic capabilities.

Additionally, our strategy used the IndoBERTFormer model, which combines a BERT encoder and a Transformer decoder. We also presented the IndoBARTFormer model, which uses a transformer for decoding vectors, like the BERT model, to broaden the scope of our research. Finally, as GPT is built on the

Table 3 Experiment Setup for Bi-LSTM and Bi-GRU Scenarios

Hyper-Parameter	Value
Epoch	15
Batch Size	64
Learning Rate	0.001
Unit (for each encoder and decoder)	256
Number of Layers (for each encoder and decoder)	2
Dropout rate	30%

Table 4 Experiment Setup for Transformer-Based Model Scenarios

Hyper-Parameter	Value
Epoch	5 to 70
Batch Size	64
Learning Rate	0.001
Feedforward Dimension	256
Number of Layers (for each encoder and decoder)	2
Dropout rate	30%
Number of Multi-head Attention	2

decoder notion, we employ the Transformer to become an encoder to improve understanding called IndoTransGPT.

4 Experimental setup

This section provides examples of our proposed model's baseline models, evaluation metrics, and hyperparameter settings.

4.1 Comparison of models

We chose a few innovative models that have received extensive research in natural language processing as baselines to show our approach's competitiveness. The following provides a succinct overview of these models' fundamental concepts. The details of hyper-parameter scenarios for every scenario are covered in Tables 3 and 4.

4.1.1 Bi-LSTM

LSTMs are recurrent neural networks (RNNs) designed to better capture the long-term dependencies in sequential data such as text. Bi-LSTM networks extend LSTMs by adding a second layer of LSTMs that process the input sequence in reverse order. This allows the network to capture both forward and backward dependencies in the

data, resulting in improved performance on tasks where understanding the context and meaning of words in a sentence is essential.

In a Bi-LSTM network, each word in a sentence is represented as an input vector and fed into the network one at a time. The network then processes the input sequence using both forward and backward LSTMs, resulting in two sets of hidden states that capture the context of the input sequence in both directions. The forward and backward hidden states are then concatenated and used as input to a subsequent network layer or output for a classification task. By stacking two BiLSTM layers as the encoder and two LSTM layers as the decoder, together with the attention mechanism suggested by Bahdanau et al. (Bahdanau et al., 2015), we create a BiLSTM model as our baseline model.

4.1.2 Bi-GRU

This problem was addressed by the GRU (Gated Recurrent Unit) development, which uses an update gate and reset gate to enable the model to store input for a longer duration and filter out irrelevant data for prediction. Two GRUs are used in a model known as BiGRU, with one accepting input going forward (forward GRU) and the other accepting input going backward (backward GRU) (backward GRU).

4.1.3 Transformer

Unlike traditional sequence-to-sequence models, which rely on recurrent neural networks (RNNs) to process sequences, Transformer uses a self-attention mechanism to weigh the importance of each element in the input sequence and generate a fixed-length representation, often referred to as the context vector or encoding.

The Transformer architecture consists of an encoder-decoder architecture, where the encoder processes the input sequence and generates the context vector. The decoder generates the output sequence based on the context vector and the previously generated output tokens. The self-attention mechanism allows the model to capture the dependencies between all elements in the sequence, making it more effective at handling long-range dependencies and reducing the reliance on fixed-length representations such as RNN hidden states. The Transformer model has become famous for many NLP applications due to its high performance, parallelizable nature, and ability to model long-range dependencies.

4.1.4 IndoBERTFormer

IndoBERT is a Transformer-based language model that uses the BERT architecture to learn contextual representations of Indonesian text. It is pre-trained on a large corpus of Indonesian text using a masked language modeling (MLM) objective and a next sentence prediction (NSP) objective. During pre-training, a random set of tokens in the input sequence is masked and the model is trained to predict the original values of these masked tokens. In addition, the model is also trained to predict whether a pair of sentences are consecutive or not, which helps it learn relationships between sentences.

IndoBERT is like other BERT models in its architecture, consisting of a multi-layer bidirectional transformer encoder that uses self-attention to process the input sequence. However, the model is adapted to the specifics of the Bahasa Indonesia, such as the use of affixes and word order.

IndoBERTFormer adapts the BERT architecture designed explicitly for the query generation task. IndoBERTFormer is distinguished from regular IndoBERT because the output is forwarded through the Transformer decoder layer. This combination is based on BERT's nature, an encoder that aims to extract context from a text. Therefore, a decoder layer is needed to generate text from the context features generated by BERT. Using pre-trained weights on IndoBERT as an encoder is expected to help the model understand the context more easily.

4.1.5 IndoTransGPT

Like previous GPT models, IndoGPT's design consists of a multi-layer transformer decoder that processes the input sequence using self-attention. The model is trained to predict the next token given the last tokens in the input sequence. As a result, the model can acquire contextual representations of the input text that accurately reflect its semantics.

A large dataset of Indonesian text, including various genres, news stories, social media postings, and literature, served as the pre-training data for IndoGPT. The model features an innovative GPT-like architecture with 12 layers, 768 hidden units, and 12 self-attention heads. In a variety of Indonesian NLP tasks, such as text classification, named entity identification, and sentiment analysis, IndoGPT has demonstrated innovative performance. Moreover, the model has been improved for various downstream tasks, including summarization and machine translation. The model may be fine-tuned on specific tasks to learn representations unique to those activities and perform better on the downstream tasks.

IndoTransGPT is a modified variant of the IndoGPT architecture designed specifically for text generation tasks. IndoTransGPT, unlike the original IndoGPT architecture, which only consisted of a decoder, integrates a Transformer encoder with the IndoGPT language model as a decoder. This combination allows the model to derive context from the input text using the Transformer encoder and generate new text using the IndoGPT language model. Replacing the generic Transformer decoder with the IndoGPT model is expected to improve the learning process because IndoGPT has already been trained with the Indonesian dataset.

4.1.6 IndoBARTFormer

The BART (Bidirectional and Auto-Regressive Transformers) architecture is the foundation for the pre-trained IndoBART language model, designed specifically for Indonesians. It was created by the Indonesian tech startup IndoNLU and trained on a large corpus of Indonesian literature to learn representations of the language that accurately reflect its semantics.

Like BART, which has an encoder-decoder architecture with a bidirectional transformer encoder and an auto-regressive transformer decoder, IndoBART has an

encoder-decoder design. While the decoder creates the output sequence one token at a time, conditionally on the preceding tokens, the encoder examines the input sequence in both ways to collect contextual information. With the help of this architecture, the model may discover contextual representations that accurately reflect the intent of the input text.

Combining autoencoding and denoising autoencoding goals allows IndoBART to be taught. In the autoencoding goal, a random subset of tokens is substituted with a unique masking token. The model is trained to recover the original input sequence from this tainted form. This goal motivates the model to acquire contextual representations that accurately reflect the input text's semantics. The model is trained to recreate the original input sequence from a sequence with random noise introduced for the denoising autoencoding aim. This goal enhances the model's capacity to oversee noisy input text.

The concept of assembling the IndoBARTFormer architecture originates from efforts to adapt the IndoBART model for the case of question creation. The decoder section, intended explicitly for text summarization, must be adapted from this trained architecture. The default IndoBART decoder was replaced with a generic Transformer decoder to suit the task in the case of this study but did not leave the IndoBART encoder pre-training weights. The IndoBARTFormer model is hoped to understand the Indonesian language context more quickly and easily.

4.2 Automated evaluation metrics

BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) are metrics used to evaluate the quality of machine-generated text in natural language processing tasks. Both BLEU and ROUGE provide a quantitative measure of how well a machine-generated text matches the human reference text, allowing researchers to objectively compare and improve the performance of their natural language processing models.

4.2.1 BLEU

Papineni et al. (2002) developed Bi-Lingual Evaluation Understudy (BLEU) approach to evaluate a translation system's effectiveness. It is based on n-gram overlap, where n-grams are contiguous sequences of n words. A good translation should have similar n-gram distributions to the reference translations.

The BLEU score is computed by counting the number of n-grams in the machine-generated translation that appear in the reference translations and then normalizing this count by the total number of n-grams in the machine-generated translation. This normalization helps to account for differences in the lengths of the translations being compared.

To address the issue of precision versus recall, the BLEU metric uses a geometric mean of the n-gram precisions, where the precision is the number of n-grams in the machine-generated translation that appear in the reference translations, divided by the total number of n-grams in the machine-generated translation. This geometric mean

encourages translations with high precision for multiple values of n rather than just for a single value of *n*.

$$p_n = \frac{\sum_{C\in\{candidates\}}\sum_{ngram\in C} Count_{clip}(ngram)}{\sum_{C\in\{candidates\}}\sum_{ngram\in C} Count(ngram)}$$

$$Count_{clip} = \min(Count, Max\_ref\_Count)$$

Calculating a shortness penalty includes stretching the candidate translation to match the length of the benchmark translation, where l_c denotes the candidate translation's length and l_r is the reference corpus's length. The BLEU score may then be obtained,

$$BP = f(x) = \begin{cases} 1, \text{if } l_c > l_r \\ e^{(1 - l_r/l_c}, \text{if } l_c \le l_c \end{cases}$$

$$BLEU = BP.\exp\left(\sum_{n=1}^{N} \frac{1}{N} \log p_n\right)$$

We used several *n* in our experiment to determine the BLEU score. Moreover, when $n>1$, we use a smoothing technique developed by Lin et al. (Lin & Och, 2004) to increase the total and hit counts of ngrams by one. In this manner, candidate translations that include fewer words than *n* might get a good grade. We utilize NLTK's version of the smoothing approach, BLEU, in our experiment.

BLEU-1 only examines unigrams and is a straightforward measure of the degree of word-for-word overlap between the output of machine translation and reference translations. In addition to unigrams, BLEU-2 and BLEU-3 consider bigrams and trigrams, respectively, to capture a higher level of syntactic similarity between the machine translation output and the reference translations. Lastly, BLEU-4 takes 4-g into account and aims to convey a higher level of semantic similarity between the output of machine translation and the reference translations. The final BLEU score is usually computed as a geometric mean of the individual BLEU-n scores.

4.2.2 ROUGE

Lin et al. (Lin, 2004) developed the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to judge the effectiveness of automatically produced summaries. The metrics ROUGEN and ROUGE-L, which we will utilize in our experiment, are among the many that it comprises. In contrast to BLEU's leaning toward *ngram* precision, ROUGE-N emphasizes *ngram* recall, which is computed as

$$\text{ROUGE-N} = \frac{\sum_{S\in\{References\}}\sum_{ngram\in C} Count_m(ngram)}{\sum_{S\in\{References\}}\sum_{ngram\in C} Count(ngram)}$$

When *ngram* indicates the reference ngrams, $Count_m$ is the maximum number of overlapping *ngram* references and candidate summaries. Briefly stated, ROUGE-N's denominator is the total number of references ngrams, and its numerator is the number of overlapping ngrams between references and candidates.

ROUGE-L employs both the Longest Common Subsequence (LCS) and the F-measure to determine the level of correspondence across two descriptions: the candidate summary S_{can} of length l_a and the reference summary S_{ref} of length l_e. According to the calculation:

$$Recall_{lcs} = \frac{LCS(S_{ref}, S_{can})}{l_e}$$

$$Precision_{lcs} = \frac{LCS(S_{ref}, S_{can})}{l_a}$$

$$ROUGE - L = \frac{Recall_{lcs} Precision_{lcs}}{Recall_{lcs} + Precision_{lcs}}$$

5 Result and discussion

According to several evaluation metrics, Tables 3, 4, and 5 compare our model's performance against other question-generating models used for SQuAD-ID, TyDiQA-ID, and IDK-MRC. We contrast our model with the above baseline techniques for all datasets from several open-source codes with certain adjustments. Our model dramatically surpasses innovative methods and performs best on both datasets. The BLEU-4 and ROUGE-L of our solution are 2.2, and 25.05 on the SQuAD-ID dataset, respectively, whereas the comparable prior state-of-the-art values are below from other methodologies. Comparing our strategy to the baselines in the table results in a much-improved performance on the dataset.

We integrate many techniques in our model to teach it when to produce the question based on the ground truth question. Then, using a Transformer-based predictor,

Table 5 The Baseline Models Utilized in the SQuAD-ID Test Set are Described in Depth

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
Bi-LSTM	20.15	4.85	0.8	0.19	23.29
Bi-GRU	19.64	4.45	0.76	0.15	22.64
Transformer	22.34	5.51	1.1	0.29	25.84
IndoBERTFormer	**22.96**	**5.93**	**1.31**	**0.37**	**26.48**
IndoTransGPT	20.99	4.93	0.89	0.22	25.05
IndoBARTFormer	17.13	3.13	0.27	0.01	24.45

our algorithm learns to foretell questions. Furthermore, our encoder includes a range of embeddings of various characteristics and hint indications. When used with the masking method, our approach can more effectively determine the connection between input and output patterns. Finally, the smaller target vocabulary makes it simpler to train the generator and helps our model better understand whether to replicate or create. Using SQuAD-ID, TyDiQA-D, and IDK-MRC, we may attain superior performance to state-of-the-art models by including new techniques and modules in our model. Our model performs best on the three datasets when all these strategies are used.

The performance on SQuAD-ID, TyDiQA-ID, and IDK-MRC differ due to the varied dataset features. According to Table 2, the average response length of TyDiQA-ID and IDK-MRC is shorter than that of SQuAD-ID. More extended responses often include more information and are more challenging to formulate questions. Due to these factors, TyDiQA-ID and IDK-MRC performance is much better than SQuAD-ID. Our technique is still notably superior to the comparable approaches for all datasets. It proves that copying from the input often occurs across various datasets. Based on our revised criteria for identifying whether a question word is copied, our model more accurately distinguishes between copied words and created words in a question.

IDK-MRC is an Indonesian Machine Reading Comprehension dataset covering answerable and unanswerable questions. The new unanswerable question in IDK-MRC is generated using a question generation model and human-written questions. On the other hand, SQuAD-id is a translation of the Stanford Question Answering Dataset (SQuAD) to Bahasa Indonesia. There could be several reasons we are seeing better results with IDK-MRC than SQuAD-id. One possibility is that IDK-MRC was explicitly designed for Indonesian Machine Reading Comprehension and includes answerable and unanswerable questions. At the same time, SQuAD-id is a translation of an English dataset.

Several factors, such as the size and quality of the datasets, the diversity of question types and topics, and the evaluation metrics used, can be why TyDiQA-ID is better than other datasets, even though with a short sequence length. Moreover, text on the TyDiQA-ID has a much lower number of tokens containing non-ASCII characters than IDK-MRC and SQuAD-ID, where these non-ASCII tokens will be converted into unknown tokens, which can reduce the model's understanding of the context. Furthermore, Moreover, the way TyDiQA and SQuAD are constructed is different. TyDiQA is based on an inclusive sampling of text from the web that covers a wide range of topics, genres, and domains. In contrast, SQuAD is based on curated articles that are specifically selected to be suitable for question-answering. This difference in dataset construction may lead to diverse types of questions being asked and different language variations in the dataset.

With ablation experiments, we assess how the various components in our model affect the system. The performance of our model versions with various sub-components eliminated is given in Tables 5, 6, and 7. The performance drastically declines when we remove the supplementary feature embeddings from our model, including POS, NER, Dependency Types, word frequency levels (low-frequent, median-frequent, high-frequent), and binary features (if it is a lowercase

Table 6 The baseline models utilized in the IDK-MRC test set are described in depth

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
Bi-LSTM	18.78	2.44	0.92	0.48	21.62
Bi-GRU	21.12	4.54	2.1	0.99	23.96
Transformer	28.65	9.32	2.62	0.92	31.79
IndoBERTFormer	**29.14**	**9.41**	**3.08**	**1.23**	**32.51**
IndoTransGPT	22.81	6.46	2.43	1.08	26.33
IndoBARTFormer	20.89	3.15	2.39	1.3	23.42

Table 7 The baseline models utilized in the TyDiQA-ID test set are described in depth

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
Bi-LSTM	18.42	2.47	0.06	0.035	21.32
Bi-GRU	22.7	5.99	0.201	0.094	26.29
Transformer	26.86	9.45	0.304	0.112	30.34
IndoBERTFormer	**30.45**	**10.39**	**0.334**	**0.149**	**34.35**
IndoTransGPT	23.35	5.31	0.06	0.018	26.85
IndoBARTFormer	23.13	3.6	0.208	0.101	25.83

digit). This is because each token is represented differently by the tags and feature embeddings. Compared to the variety of words, there are far less diverse tags. Hence, the patterns discovered from these tags and characteristics are more apparent than what we can learn from word embeddings. Explicitly concatenating these feature embedding vectors allows the model to catch the patterns needed to pose a question more readily, even when a well-trained word vector already has information about other features like POS or NER.

The performance of our model suffers when the target vocabulary reduction procedure is disabled. As previously noted, high-frequency words encompass nonoverlap inquiry words (or created words). Our model learns the probability of creating these terms more effectively when the amount of the target vocabulary is decreased. However, it also motivates the model to better replicate what it can from the input text.

Lastly, performance on three datasets needs the self-made word corpus prediction module. This is because asking a question about a passage's answer span still involves a one-to-many mapping difficulty. Our self-made word corpus prediction module learns how people choose the corresponding self-made word corpus by learning from a sizable training dataset to minimize the ambiguity of how to pose a question. Our model can fit the questions asked in the dataset by using projected clue indicators as part of the generator's encoder. Tables 8, 9, and 10 contain some examples of the questions that each model on the SQuAD-ID, IDK-MRC, and TyDiQA-ID datasets generated in Indonesian. The context

Table 8 Example predictions for AQG task for all models SQuAD-ID

Sample Prediction – SQuAD-ID		English Translation
Paragraph	Pusat kota San Diego adalah kawasan pusat bisnis San Diego, meskipun kota ini dipenuhi dengan kawasan bisnis	Downtown San Diego is San Diego's central business district, although the city is filled with business districts
Answer	San Diego	San Diego
Target Question	Apa distrik pusat bisnis di pusat kota San Diego?	What is the central business district in downtown San Diego?
Bi-LSTM	Apa pusat ritel terbesar di San Diego?	What is the largest retail center in San Diego?
Bi-GRU	Kota mana yang paling dikenal sebagai Kota Boston?	Which city is best known as the City of Boston?
Transformer	Apa nama perusahaan pemula yang menawarkan bisnis di San Diego?	What is the name of a start-up company offering business in San Diego?
IndoBERTFormer	Apa pusat kota San Diego?	What is downtown San Diego?
IndoTransGPT	Apa nama kano yang diberikan kepada San Diego?	What was the name of the canoe given to San Diego?
IndoBARTFormer	Apa yang tidak diposting?	What's not posted?

Table 9 Example predictions for AQG task for all models IDK-MRC

Sample Prediction – IDK-MRC		English Translation
Input Sentence & Answer	Judul kakawin ini, Nagarakretagama artinya adalah "Negara dengan Tradisi (Agama) yang suci"	The title of this kakawin, Nagarakretagama means "Country with Holy (Religious) Traditions"
Answer	Negara dengan Tradisi (Agama) yang suci	Country with Holy (Religious) Traditions
Target Question	Apa yang dimaksud dengan Nagarakretagama?	What is meant by Nagarakretagama?
Bi-LSTM	Kapan istilah Belanja daring?	When is the term Online shopping?
Bi-GRU	Apakah jabatan terakhir dalam bidang politik?	What was your last position in politics?
Transformer	Apa itu wiracarita Mahabharata?	What is the epic Mahabharata?
IndoBERTFormer	Apa yang dimaksud dengan karya Mahabharata?	What is meant by the Mahabharata?
IndoTransGPT	Apa yang dimaksud dengan zaman ekonomi?	What is meant by the economic age?
IndoBARTFormer	Apa yang dimaksud dengan yang tidak?	What is meant by not?

Table 10 Example predictions for AQG task for all models TyDiQA-ID

Sample Prediction—TyDiQA-ID		English Translation
Input Sentence & Answer	Ibu kotanya, Sri Jayawardenapura Kotte, adalah kota penggiran dari kota terbesar di Sri Lanka, Kolombo	The capital, Sri Jayawardenapura Kotte, is a suburb of Sri Lanka's largest city, Colombo
Answer	Sri Jayawardenapura Kotte	Sri Jayawardenapura Kotte
Target Question	Apakah ibukota Sri Lanka?	What is the capital of Sri Lanka?
Bi-LSTM	Apakah nama ibukota Belanda?	What is the name of the capital of the Netherlands?
Bi-GRU	apakah nama ibukota Skotlandia?	what is the name of the capital of Scotland?
Transformer	Apakah ibukota Sulawesi?	What is the capital of Sulawesi?
IndoBERTFormer	Apakah ibukota Sri Lanka?	What is the capital of Sri Lanka?
IndoTransGPT	Dimana letak kota Sulawesi?	Where is the city of Sulawesi located?
IndoBARTFormer	Apa nama ibukota yang di maksud dengan negara?	What is the name of the capital which is meant by the state?

or passage used as the model input is referred to as "Input Sentence & Answer," followed by the anticipated response. The "Target Question" is the anticipated produced question.

6 Conclusion

This study investigates IndoBERTFormer and IndoBARTFomer, whose decoders use Transformer layers and are trained using a word-based coverage approach. Apart from that, we also explored IndoTransGPT using a generic Transformer as the encoder. Our methodology efficiently uses context-to-answer attention more reliably than longer answers to extract more relevant information from surrounding sentences.

The main contribution of this paper is to compare several state-of-the-art pre-trained models to create an automatic question generator with narrative paragraphs as input. This model is intended to assist teachers in creating short answer questions. It also provides additional options for students to study using the practice question method. This paper also makes it easier for AI developers to choose which pre-trained models are most effective and efficient for cases like those that form the background of this research. Our experimental results show that IndoBERTFormer has the best accuracy, fluency, and diversity of questions generated.

The scope and limitations of this research are as follows: First, this research only evaluates these models for the case of creating short answer questions with input data in the form of descriptive paragraphs of 100 to 500 words. This research has yet to evaluate these models for multiple-choice or other types of questions. Second, this research only uses input text in Indonesian. This research has yet to evaluate these models for processing input text other than Indonesian or multilingual. Third, this research only uses three trained models: IndoBERTFormer, IndoBARTFomer, IndoTransGPT, and several other comparison models. This research has yet to evaluate other models for which pre-trained weights from the Indonesian dataset are available or compared with conventional machine learning methods. This research investigates IndoBERTFormer and IndoBARTFomer, whose decoders use Transformer layers and are trained using a word-based approach coverage. Apart from that, we also explored IndoTransGPT using a generic Transformer as the encoder. Our methodology efficiently uses context-to-answer attention more reliably than longer answers to extract more relevant information from surrounding sentences.

As part of our expanded response, we want to investigate more sophisticated and trainable techniques, such as multitasking and semi-supervised learning, in future work. Moreover, we would like to investigate how well the pre-trained model performs in additional language creation tasks that profit from the copy mechanism, such as text summarization and neural machine translation.

Appendix

Example Predictions for AQG Task for All Models SQuAD-ID

Sample Prediction – SQuAD-ID		English Translated	Evaluation
Paragraph	Masalah fungsi adalah masalah komputasi di mana output tunggal (dari fungsi total) diharapkan untuk setiap input, tetapi output lebih kompleks daripada masalah keputusan, yaitu, itu bukan hanya ya atau tidak	A function problem is a computational problem in which a single output (of the total function) is expected for each input, but the output is more complex than a decision problem, that is, it's not just a yes or no	
Answer	masalah komputasi	computational problem	
Target Question	Apa yang dimaksud dengan solusi fungsi?	What is a functional solution?	
Bi-LSTM	Jenis masalah apa yang bisa diukur dengan asma?	What types of problems can asthma measure?	BLEU 1: 33.33 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 38.36 BLEU AVG: 8.3325
Bi-GRU	Apa salah satu hal utama pengirim dan informasi yang ditentukan secara ekonomis untuk meningkatkan hasil akhir?	What is one of the main things shipping and information determines economically to improve the end result?	BLEU 1: 17.65 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 27.03 BLEU AVG: 4.4125
Transformer	Apa yang bisa menyebabkan kerusakan?	What could be causing the damage?	BLEU 1: 42.32 BLEU 2: 26.77 BLEU 3: 0 BLEU 4: 0 ROUGE L: 45.52 BLEU AVG: 17.2725
IndoBERTFormer	Apa yang dimaksud dengan DBMS biner?	What is a binary DBMS?	BLEU 1: 71.43 BLEU 2: 59.76 BLEU 3: 52.28 BLEU 4: 43.47 ROUGE L: 71.43 BLEU AVG: 56.735
IndoTransGPT	Apa yang bisa digunakan untuk mengukur apakah refleksi?	What can be used to measure whether reflection?	BLEU 1: 33.33 BLEU 2: 20.41 BLEU 3: 0 BLEU 4: 0 ROUGE L: 38.36 BLEU_AVG: 13.435
IndoBARTFormer	Apa yang tidak diposting?	What's not posted?	BLEU 1: 40.22 BLEU 2: 25.96 BLEU 3: 0 BLEU 4: 0 ROUGE L: 48.54 BLEU AVG: 16.544

Example Predictions for AQG Task for All Models IDK-MRC

Sample Prediction – IDK-MRC		English Translated	Evaluation
Paragraph	Luas wilayah Dinasti Qing pada masa puncaknya pernah mencapai 12 juta kilometer persegi	The area of the Qing Dynasty at its peak once reached 12 million square kilometers	
Answer	12 juta kilometer persegi	12 million square kilometers	
Target Question	berapakah luas kekuasaan Dinasti Qing?	How large was the Qing dynasty?	
Bi-LSTM	Berapa luas Pulau Misool?	How wide is Misool Island?	BLEU 1: 32.75 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 35.78 BLEU AVG: 8.1875
Bi-GRU	Berapa luas kota Sawahl-unto?	How wide is the city of Sawahlunto?	BLEU 1: 32.75 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 35.78 BLEU AVG: 8.1875
Transformer	Berapa luas wilayah Dinasti Tang?	How large is the territory of the Tang Dynasty?	BLEU 1: 50 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 50 BLEU AVG: 12.5
IndoBERTFormer	Berapa luas kekuasaan Dinasti Tang?	How large is the Tang dynasty?	BLEU 1: 66.67 BLEU 2: 51.64 BLEU 3: 40.55 BLEU 4: 0.01 ROUGE L: 66.67 BLEU AVG: 39.7175
IndoTransGPT	Berapa luas negara yang dimaksud dengan Den-mark?	What is the area of the coun-try is Denmark?	BLEU 1: 25 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 29.33 BLEU AVG: 6.25
IndoBARTFormer	Apa yang dimaksud dengan zaman?	What is meant by age?	BLEU 1: 16.67 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE_L: 16.67 BLEU AVG: 4.1675

Example Predictions for AQG Task for All Models TyDiQA-ID

Sample Prediction – TyDiQA-ID		English Translation	Evaluation
Paragraph	Istilah "mitologi" dapat berarti kajian tentang mitos (misalnya mitologi perbandingan), maupun sebuah himpunan atau koleksi mitos-mitos (misalnya mitologi Inka)	The term "mythology" can mean the study of myths (e.g., comparative mythology), as well as a set or collection of myths (e.g., Incan mythology)	
Answer	kajian tentang mitos (misalnya mitologi perbandingan), maupun sebuah himpunan atau koleksi mitos-mitos	the study of myths (e.g., comparative mythology), as well as a set or collection of myths	
Target Question	Apa itu mitologi?	What is mythology?	
Bi-LSTM	Apa itu media massa?	What is mass media?	BLEU 1: 60 BLEU 2: 38.73 BLEU 3: 0 BLEU 4: 0 ROUGE L: 68.03 BLEU AVG: 24.6825
Bi-GRU	Apa itu algoritme?	What is an algorithm?	BLEU 1: 75 BLEU 2: 50 BLEU 3: 0 BLEU 4: 0 ROUGE L: 75 BLEU AVG: 31.25
Transformer	Apakah yang dimaksud dengan mitologi?	What is meant by mythology?	BLEU 1: 33.33 BLEU 2: 25.82 BLEU 3: 0 BLEU 4: 0 ROUGE L: 41.5 BLEU AVG: 14.7874
IndoBERTFormer	Apa itu mitologi?	What is mythology?	BLEU 1: 100 BLEU 2: 100 BLEU 3: 100 BLEU 4: 100 ROUGE L: 100 BLEU AVG: 100
IndoTransGPT	Apa itu dimaksud dengan molekul?	What is meant by a molecule?	BLEU 1: 50 BLEU 2: 31.62 BLEU 3: 0 BLEU 4: 0 ROUGE L: 62.24 BLEU AVG: 20.405
IndoBARTFormer	Apa yang dimaksud dengan teori?	What is meant by theory?	BLEU 1: 33.33 BLEU 2: 0 BLEU 3: 0 BLEU 4: 0 ROUGE L: 41.5 BLEU AVG: 8.3325

Data availability The datasets used in this study, namely SQuAD-ID, IDK-MRC, and TyDiQA-ID, can be freely accessed on GitHub (https://github.com/IndoNLP/nusa-crowd). Researchers and interested parties can get these datasets from the specified source to replicate, do further analysis, or construct models. We promote unrestricted access to these resources to promote transparency, collaboration, and advancement within the natural language processing community.

Declarations

Ethics approval Ethical clearance is not applicable.

Conflict of interest The authors declare no conflicts of interest.

References

Abosalem, Y. (2015). Assessment techniques and students' higher-order thinking skills. *ICSIT 2018 - 9th International Conference on Society and Information Technologies, Proceedings*, *4*(1), 61–66. https://doi.org/10.11648/j.ijsedu.20160401.11

Akyön, F. Ç., Çavuşoğlu, D., Cengiz, C., Altinuç, S. O., & Temizel, A. (2022). Automated question generation and question answering from Turkish texts. *Turkish Journal of Electrical Engineering and Computer Sciences,30*(5), 1931–1940. https://doi.org/10.55730/1300-0632.3914

Al-Chalabi, H. K. M., Hussein, A. M. A., & Apoki, U. C. (2021). An adaptive learning system based on learner's knowledge level. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*, *13*(12), 191–200. https://doi.org/10.1109/ECAI52376.2021.9515158

Almaiah, M. A., & Al Mulhem, A. (2019). Analysis of the essential factors affecting of intention to use of mobile learning applications: A comparison between universities adopters and non-adopters. *Education and Information Technologies,24*(2), 1433–1468. https://doi.org/10.1007/s10639-018-9840-1

Almaiah, M. A., & Jalil, M. A. (2014). Investigating students' perceptions on mobile learning services. *International Journal of Interactive Mobile Technologies,8*(4), 31–36. https://doi.org/10.3991/ijim.v8i4.3965

Almaiah, M., Jalil, M. A., & Man, M. (2016). Preliminary study for exploring the major problems and activities of mobile learning system: A case study of JORDAN. *Journal of Theoretical and Applied Information Technology*, *93*(2). http://www.jatit.org. Accessed 28 Mar 2023.

Almaiah, M. A., Alamri, M. M., & Al-Rahmi, W. M. (2020). Analysis the effect of different factors on the development of mobile learning applications at different stages of usage. *IEEE Access,8*, 16139–16154. https://doi.org/10.1109/ACCESS.2019.2963333

Almaiah, M. A., Al-Khasawneh, A., Althunibat, A., & Almomani, O. (2021). *Exploring the Main Determinants of Mobile Learning Application Usage During Covid-19 Pandemic in Jordanian Universities* (pp. 275–290). https://doi.org/10.1007/978-3-030-67716-9_17

Almaiah, M. A., Ayouni, S., Hajjej, F., Lutfi, A., Almomani, O., & Awad, A. B. (2022). Smart mobile learning success model for higher educational institutions in the context of the COVID-19 pandemic. *Electronics,11*(8), 1278. https://doi.org/10.3390/electronics11081278

Alsubait, T., Parsia, B., & Sattler, U. (2016). Ontology-based multiple choice question generation. *KI - Künstliche Intelligenz,30*(2), 183–188. https://doi.org/10.1007/s13218-015-0405-9

Annamoradnejad, I., Fazli, M., & Habibi, J. (2020). Predicting Subjective Features from Questions on QA Websites using BERT. *2020 6th International Conference on Web Research (ICWR)*, pp. 240–244. https://doi.org/10.1109/ICWR49608.2020.9122318

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15.

Blegur, J., Rajagukguk, C. P. M., Sjioen, A. E., & Souisa, M. (2023). Innovation of analytical thinking skills instrument for throwing and catching game activities for elementary school students. *International Journal of Instruction,16*(1), 723–740.

Bordes, A., Usunier, N., Chopra, S., & Weston, J. (2015). Large-scale Simple Question Answering with Memory Networks. *CoRR*, *abs/1506.0*.

Cahyawijaya, S., Winata, G. I., Wilie, B., Vincentio, K., Li, X., Kuncoro, A., Ruder, S., Lim, Z. Y., Bahar, S., Khodra, M. L., Purwarianti, A., & Fung, P. (2021). IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 8875–8898. https://doi.org/10.18653/v1/2021.emnlp-main.699

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). {S}em{E}val-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation ({S}em{E}val-2017)*, pp. 1–14. https://doi.org/10.18653/v1/S17-2001

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using {RNN} Encoder-Decoder for Statistical Machine Translation. *CoRR*, *abs/1406.1*.

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., & Palomaki, J. (2020). TyDiQA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics,8*, 454–470. https://doi.org/10.1162/tacl_a_00317

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). {BERT}: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., & Hon, H.-W. (2019). Unified Language Model Pre-training for Natural Language Understanding and Generation. *CoRR*, *abs/1905.0*.

Du, X., Shao, J., & Cardie, C. (2017). Learning to Ask: Neural Question Generation for Reading Comprehension. *CoRR*, *abs/1705.0*.

Garneau, N., Leboeuf, J.-S., & Lamontagne, L. (2019). Predicting and interpreting embeddings for out of vocabulary words in downstream tasks. *CoRR*, *abs/1903.0*.

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks. *International Conference on Machine Learning*, 1319–1327.

Hao, T., Li, X., He, Y., Wang, F. L., & Qu, Y. (2022). Recent progress in leveraging deep learning methods for question answering. *Neural Computing and Applications,34*(4), 2765–2783. https://doi.org/10.1007/s00521-021-06748-3

Harrison, V., & Walker, M. (2018). Neural Generation of Diverse Questions using Answer Focus, Contextual and Linguistic Features. *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 296–306. https://doi.org/10.18653/v1/W18-6536

Hunter, A., Chalaguine, L., Czernuszenko, T., Hadoux, E., & Polberg, S. (2019). *Towards Computational Persuasion via Natural Language Argumentation Dialogues BT - KI 2019: Advances in Artificial Intelligence* (C. Benzmüller & H. Stuckenschmidt, Eds.; pp. 18–33). Springer International Publishing.

Jurafsky, D. (2000). *Speech and language processing*. Pearson Education India.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications,82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. *ArXiv Preprint. ArXiv:2011.00677*.

Kumar, V., Ramakrishnan, G., & Li, Y.-F. (2018). A framework for automatic question generation from text using deep reinforcement learning. *CoRR*, *abs/1808.0*.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education,30*(1), 121–204. https://doi.org/10.1007/s40593-019-00186-y

Kusuma, S. F., Siahaan, D. O., & Fatichah, C. (2022). Automatic question generation with various difficulty levels based on knowledge ontology using a query template. *Knowledge-Based Systems,249*, 108906. https://doi.org/10.1016/j.knosys.2022.108906

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 74–81.

Lin, C.-Y., & Och, F. J. (2004). Orange: a method for evaluating automatic evaluation metrics for machine translation. *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pp. 501–507.

Liu, B., Lai, K., Zhao, M., He, Y., Xu, Y., Niu, D., & Wei, H. (2019a). Learning to generate questions by learning what not to generate. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pp. 1106–1118. https://doi.org/10.1145/3308558.3313737

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., & Smith, N. A. (2019b). Linguistic Knowledge and Transferability of Contextual Representations. *Proceedings of the 2019 Conference of the North {A} merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1073–1094. https://doi.org/10.18653/v1/N19-1112

Luong, T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421. https://doi.org/10.18653/v1/D15-1166

Mazidi, K., & Tarau, P. (2016). *Automatic Question Generation: From NLU to NLG BT - Intelligent Tutoring Systems* (A. Micarelli, J. Stamper, & K. Panourgia, Eds.; pp. 23–33). Springer International Publishing.

Muis, F. J., & Purwarianti, A. (2020). Sequence-to-Sequence Learning for Indonesian Automatic Question Generator. *2020 7th International Conference on Advanced Informatics: Concepts, Theory and Applications, ICAICTA 2020*. https://doi.org/10.1109/ICAICTA49861.2020.9429032

Palvia, S., Aeron, P., Gupta, P., Mahapatra, D., Parida, R., Rosner, R., & Sindhi, S. (2018). Online education: Worldwide status, challenges, trends, and implications. *Journal of Global Information Technology Management,21*(4), 233–241. https://doi.org/10.1080/1097198X.2018.1542262

Papasalouros, A., & Chatzigiannakou, M. (2018). Semantic Web and Question Generation: An Overview of the State of the Art. *International Association for Development of the Information Society*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Putri, R. A., & Oh, A. (2022). *IDK-MRC: Unanswerable Questions for Indonesian Machine Reading Comprehension*.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog,1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research,21*(1), 5485–5551.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuad: 100,000+ questions for machine comprehension of text. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2383–2392. https://doi.org/10.18653/v1/d16-1264

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *2*, 784–789. https://doi.org/10.18653/v1/p18-2124

Rogers, A., Gardner, M., & Augenstein, I. (2023). Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys,55*(10), 1–45.

Salazar, J., Liang, D., Nguyen, T. Q., & Kirchhoff, K. (2020). *Masked Language Model Scoring*. *Figure 1*, 2699–2712. https://doi.org/10.18653/v1/2020.acl-main.240

See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. *CoRR*, *abs/1704.0*.

Serban, I. V., Garc\'\ia-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., & Bengio, Y. (2016). Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer

Corpus. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 588–598. https://doi.org/10.18653/v1/P16-1056

Shigehalli, P. R. (2020). *Natural language understanding in argumentative dialogue systems*.

Sundermeyer, M., Alkhouli, T., Wuebker, J., & Ney, H. (2014). Translation modeling with bidirectional recurrent neural networks human language technology and pattern recognition group. *Emnlp, 2014*, 14–25.

Vaswani, A., Shazeer, N., & Parmar, N. (2017). Attention Is All You Need. *31st Conference on Neural Information Processing Systems (NIPS)*, *8*(1), 8–15. https://doi.org/10.1109/2943.974352

Vie, J.-J., Popineau, F., Bruillard, É., & Bourda, Y. (2017). A review of recent advances in adaptive assessment. *Learning Analytics: Fundaments, Applications, and Trends: A View of the Current State of the Art to Enhance e-Learning*, 113–142.

Vincentio, K., & Suhartono, D. (2022). Automatic question generation monolingual multilingual pre-trained models using RNN and transformer in low resource Indonesian language. *Informatica,46*(7), 103–118. https://doi.org/10.31449/inf.v46i7.4236

Yao, L., & Guan, Y. (2019). An Improved LSTM Structure for Natural Language Processing. *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018*, pp. 565–569. https://doi.org/10.1109/IICSPI.2018.8690387

Zhang, S., Zhang, X., Wang, H., Cheng, J., Li, P., & Ding, Z. (2017). Chinese medical question answer matching using end-to-end character-level multi-scale CNNs. *Applied Sciences (Switzerland),7*(8), 1–17. https://doi.org/10.3390/app7080767

Authors and Affiliations

Derwin Suhartono[1] · Muhammad Rizki Nur Majiid[1] · Renaldy Fredyan[1]

✉ Derwin Suhartono
dsuhartono@binus.edu

Muhammad Rizki Nur Majiid
muhammad.majiid@binus.ac.id

Renaldy Fredyan
renaldy.fredyan@binus.ac.id

[1] Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com