



A question–answer generation system for an asynchronous distance learning platform

Hei-Chia Wang^{1,2} · Martinus Maslim^{1,3} · Chia-Hao Kan¹

Received: 30 October 2022 / Accepted: 20 February 2023 / Published online: 4 March 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Distance learning frees the learning process from spatial constraints. Each mode of distance learning, including synchronous and asynchronous learning, has disadvantages. In synchronous learning, students have network bandwidth and noise concerns, but in asynchronous learning, they have fewer opportunities for engagement, such as asking questions. The difficulties associated with asynchronous learning make it difficult for teachers to determine whether students comprehend the course material. Motivated students will consistently participate in a course and prepare for classroom activities if teachers ask questions and communicate with them during class. As an aid to distance education, we want to automatically generate a sequence of questions based on asynchronous learning content. In this study, we will also generate multiple-choice questions for students to answer and teachers to easily correct. The asynchronous distance teaching-question generation (ADT-QG) model, which includes Sentences-BERT (SBERT) in the model architecture to generate questions from sentences with a higher degree of similarity, is proposed in this work. With the Wiki corpus generation option, it is anticipated that the Transfer Text-to-Text Transformer (T5) model will generate more fluent questions and be more aligned with the instructional topic. The results indicate that the questions created by the ADT-QG model suggested in this work have good fluency and clarity indicators, showing that the questions generated by the ADT-QG model are of a certain quality and relevant to the curriculum.

Keywords Distance learning · Question generation · T5 · Sentences-BERT (SBERT)

✉ Hei-Chia Wang
hewang@mail.ncku.edu.tw

Extended author information available on the last page of the article

1 Introduction

Due to the prevalence and adaptability of the internet, conventional face-to-face education has also experienced significant changes (Liu et al., 2019). Recent changes may be seen in the extensive usage of distance learning. It appears that we are undergoing an era of transformation in education, a move from the old, face-to-face teaching model to new teaching and learning models that employ contemporary pedagogical approaches that make use of technological advances and respond to current societal requirements (Zagouras et al, 2022). In addition, the COVID-19 pandemic has challenged education systems around the world, forcing educators to switch to distance learning models overnight (Dhawan, 2020; Gamage et al, 2022). Distance learning enables students to study online without being physically present in the classroom, and in this method, educational institutions develop and prepare instructional materials (Kaplan & Haenlein, 2016). The concept of distance learning breaks the limitation of spatial constraints in the learning process. Distance learning has been suggested for more than a century, and the medium has changed from non-electronic communication at the beginning (Spector et al., 2014) to online courses today. Several types of technological resources are available for distance learning, including audio podcasts, videos, various simulators, and online quizzes (Masalimova, et al, 2022). The benefit of these resources is that students can access and reuse them (Martha et al., 2021; Önöral & Kurtulmus-Yilmaz, 2020).

Distance learning and distance teaching are further divided into synchronous and asynchronous types. This difference refers to the time and location of teaching and learning activities (Fabrız et al., 2021). Synchronous distance teaching uses a video system so that students can watch the teachers on the other end, and the teacher also interacts with the students through video (Hrastinski, 2008). Synchronous learning provides a live platform that facilitates more direct connections and immediate reactions between instructors and students via audio-conferencing (e.g., online phone conversations and web chats) or video-conferencing applications (e.g., Zoom, Microsoft Teams, Skype, and Tencent Meeting) (Zhang & Wu, 2022). In synchronous courses, students engage in interactive and focused activities that help them gain a foundational understanding of technology-enhanced education, course design, and effective online teaching (Debes, 2021). Real-time interpersonal connection, the use of human speech, and instant feedback are the primary benefits of synchronous distance learning (Blau et al., 2017). However, there are disadvantages to synchronous distance learning. Although there are no geographical restrictions for synchronous distance education, there will be network bandwidth and noise issues (Belt & Lowenthal, 2022; Perveen, 2016). Teachers may utilize the idea of asynchronous distance learning to address the issues of synchronous distance learning. In asynchronous distance teaching, teachers record class files in advance and provide students with a platform on which to study (Hrastinski, 2008). Asynchronous learning, which can be assisted by media such as emails, forums, blogs, and previously recorded videos, is characterized by glaring time gaps between transmitters and receivers. Learners can

access the learning resources at any time and can devote extra time to contemplating problems or polishing their contributions (Zhang & Wu, 2022). However, asynchronous distance teaching still faces problems. Many teachers report that students have poor concentration while learning (Lemay et al., 2021). Additionally, the students have less interaction, such as asking questions, when asynchronous learning is used. It is difficult for the teacher to know whether the students understand the teaching content or are paying attention to the course. In this condition, students may struggle to stay focused. There is research that notes that the concentration of students may drop after 20–30 min (d’Inverno et al., 2003). The challenge is how to help teachers ensure student concentration during learning. This is a very important issue, especially when the course is online.

Lin (2015) pointed out that if teachers ask questions and interact with students in the classroom, motivated learners will regularly participate in the course and prepare for the classroom activities. This causes students to actively participate in the classroom and persist through challenges. Based on this idea, we aim to solve the problem by automatically generating a series of questions based on the content of asynchronous distance teaching as a tool to assist distance teaching. The proposed method should detect whether students understand what they have just learned and improve students’ concentration through automatic questioning without increasing the burden on the teachers to create questions. In addition, for the convenience of students in answering questions and easier correction by teachers, the method of this study will also generate multiple-choice questions. Multiple choice questions (MCQs) are a popular method of evaluation, in which respondents are asked to choose the best response from a list of options (Patra & Saha, 2018). With this type of question, students will be able to answer in real time and speed up their thinking process. We also examine the issue of online cheating; questions will be produced at random, preventing students from discussing the answers.

Some proposed question generation methods (Belkin et al., 2019; Xu et al., 2020) used deep neural network models, but deep neural networks usually have a large number of parameters and insufficient training data, which may cause overfitting. Additionally, the calculation time is very long. Pretraining problem generation could be the solution, as it can be done through pretraining language models. Only a small number of samples need to be used to generate questions, and the model does not need to be retrained. The computing time is reasonably short, such as in Bidirectional Encoder Representations from Transformers (BERT) or T5 models, for both semantic understanding and text generation tasks. In particular, T5 proposes a general framework that converts all tasks into a standard form, and it can use various materials to perform text-to-text tasks. Consequently, T5 will be used for processing question generation in this study.

To obtain the key points of the teaching materials with greater diversity, the method refers to the content of the teaching lecture as part of question generation. Additionally, we use the Google Cloud Speech-to-Text API provided by Google to extract the words that the teacher says in class. Roh and Lee (2017) noted that the Google Cloud Speech API has a good overall understanding of sentences and has the best recognition rate for standard languages. However, the lecture content may include many words that are irrelevant to the classroom, which will mean that the

generated questions do not meet the needs of the classroom. Therefore, the method needs to filter out sentences based on the lecture content through the calculation of sentence similarity to obtain the sentences that are most closely related to the class content; most previous research used methods such as Term Frequency–Inverse Document Frequency (TF-IDF), Word Representations in Vector Space (Word2Vec), or simply Global Vector (GloVe) to convert sentences into sentence vectors. Then, the sentence phasors of multiple sentences were used to calculate the similarity. However, the above method only converts the vector according to the number of occurrences in the text and does not take into account the meaning of very many contextual words or shallowness. This study will use the Sentence-BERT proposed by Reimers and Gurevych (2019) to retain the semantic meanings of sentences.

Overall, this study proposes a pretraining question generation method to assist distance teaching. The spoken content of the class is used to generate relevant questions. In the sentence selection step, Sentence-BERT first generates the sentence vector of the sentence, filters the content of the teacher’s lecture through text similarity calculation, and extracts the sentences that are important in generating questions. The pretrained answer retrieval model obtains answers and sentences, and Word2Vec generates multiple-choice options through the Wiki corpus. In the question construction step, the T5-based pretraining method is also used to generate questions.

2 Literature review

2.1 Question generation

Question generation refers to the use of natural language processing technology to generate a corresponding question for a given sentence or paragraph. Question generation can be applied in many fields, such as education. A question generation example is given in Fig. 1. The blue and brown blocks are the generated target answers A, and the questions generated for the target answers are Q. Mitkov (2003) proposed a system to generate multiple-choice questions from e-learning archives using a variety of natural language processing techniques. Mostow and Jang (2012)

Context: In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. The main forms of precipitation include drizzle, rain, sleet, snow and hail.

QG: What causes precipitation to fall?

QG: What is another main form of precipitation besides drizzle, rain, sleet and hail?

Fig. 1 Generating Question Examples Using Predicted Answers

also proposed a system for generating multiple-choice questions from e-learning archives. The last word in the paragraph is removed to generate fill-in-the-blank questions with options. The methods of question generation can be mainly divided into three categories. The first category is automatic question generation (AQG), which first converts sentences into syntactic representations and then generates questions through manually constructed question templates. The second is neural question generation (NQG). A neural network is used to extract the target answer from a given article or paragraph, and then the corresponding question is generated according to the target answer. The third category is pretraining question generation. Through transfer learning, a few samples are used to generate questions.

Among these methods, pretraining problem generation is currently mainstream. The most important aspect of pretraining problem generation is adding the concept of transfer learning. By introducing transfer learning, it is possible to solve new problems with very few samples instead of using much data to train new models from scratch. Transfer learning has a two-stage learning framework. During the pretraining stage, knowledge is acquired from one or more tasks. Next, in the fine-tuning stage, the acquired knowledge is transferred to the target task. Because of the rich knowledge acquired in the pretraining stage, the fine-tuning stage can enable the model to handle tasks with a limited number of samples. Chan and Fan (2019) added BERT for training in the process of question generation for the first time. Research has proven that the questions generated by the model proposed by scholars are semantically smoother than those generated by a recurrent neural network (RNN) and can better deal with long-term dependency problems.

2.2 Word embedding

Word embedding is a method of text vectorization that can map words into a low-dimensional real vector space and generate a representation vector representing the semantics of each word. Most of the traditional word embedding models were developed based on the concept of context-independent representation. At present, the more common methods include Word2vec, proposed by Mikolov et al. (2013), and Global Vector (GloVe), proposed by Pennington et al. (2014). In this study, the computation time is an important issue. Word2vec has the advantages of simple concepts and less computing space, and the selection of the training model generated by the multiple-choice options will be implemented using Word2vec and the English Wiki corpus.

The above methods only use a set of fixed vectors to express the semantics of each word. However, in actual cases, the same word often has a completely different tone and semantics in different contexts. Therefore, the method of context-aware representation is considered. The best-known representation is the BERT model proposed by Devlin et al. (2019). The BERT framework includes two stages: pretraining and fine-tuning. In the pretraining stage, the model is pre-trained with a large amount of unlabeled text data. In the fine-tuning phase, the model is initialized with pretrained parameters, and then all parameters are fine-tuned using task-specific labeled data. This research demonstrated that pretrained

representations reduce the task-specific architecture requirements, and BERT outperforms many task-specific architectures on sentence-level tasks.

In the similarity processing step, because of the success of BERT, Reimers and Gurevych (2019) proposed the Sentence BERT method, which averages the vectors in the BERT output layer, greatly reducing the training dimension, and uses the twin (Siamese) and three-level (triplet) network structure to obtain semantically meaningful sentence vectors; semantically similar sentences have relatively close vector-vector distances, so cosine similarity or the Manhattan or Euclidean distance can be used to find semantically similar sentences.

2.3 Deep learning

Deep learning is a neural network method in which a computational model composed of multiple processing layers performs representational learning of data. It has produced good results in many fields. In recent years, research in natural language processing has increasingly focused on the use of new deep learning methods. At present, the most commonly used pretraining models for generation tasks are BART and T5 (Zhou et al., 2020). T5 abbreviates Transfer Text-To-Text Transformer. “Transfer” comes from transfer learning, and “Text-to-Text” is a unified framework proposed by the author. By relying on a large amount of data, all neuro-linguistic programming tasks are converted into text-to-text tasks. All neuro-linguistic programming tasks can be performed with the same model, the same loss function, the same training process, and the same decoding process. In the unsupervised learning process, the same form of BERT is used. Some of the words in the sentence are MASKed and predicted. In the MASK step, the replace span (small paragraph replacement) method is used, which replaces each MASKed term with the [M] symbol to improve the calculation efficiency. The percentage of MASKed terms was tested by the author, and 15% was chosen as the appropriate value. The above strategies and larger datasets enable T5 to perform better in many natural language understanding tasks, and in text generation tasks, it is even better than the currently existing models, showing the applicability of T5 in text generation tasks.

3 Research methods

3.1 Framework

This research presents a method for generating auxiliary questions for distance teaching with three modules. The first module includes a voice filtering method based on a pretraining model. The second and third modules adapt the data from the first module to build a pretraining answer-question generation method. The framework is shown in Fig. 2. Each module is described in the following sections.

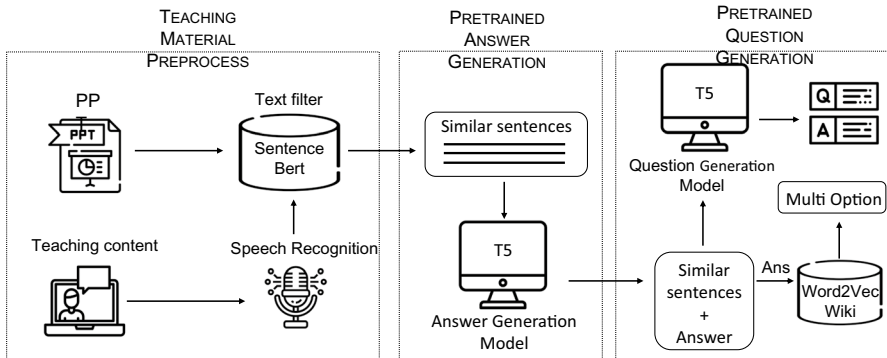


Fig. 2 Method Framework

3.2 Teaching material preprocessing module

This study utilizes the teaching materials, including teaching slides and audio files of the class, used in question generation for a distance teaching course. In the data preprocessing stage, the collected course materials are preprocessed separately for subsequent comparison of text-to-text mapping and generation of test questions.

3.2.1 Speech recognition module

One important issue for lecture review questions is to obtain the content taught by the teacher. We add the content of the teacher's lecture as part of the question generation and use the Google Cloud Speech-to-Text API to perform speech-to-text conversion. In speech recognition processing, the following two steps are needed: (1) Video-to-Audio Conversion. First, the asynchronous class has a recorded voice in the video, and it can be captured and converted into a WAV audio file format for subsequent voice recognition. (2) Speech Recognition. The audio-to-text converted file of slide page k is represented as Au_k , where $\{Au_1, Au_2, \dots, Au_k, \dots\}$ records the teacher's words. Assuming m sentences are spoken to explain slide k , the sentences described by Au_k can be represented as $\{Au_Sentence_{k1}, Au_Sentence_{k2}, \dots, Au_Sentence_{km}\}$.

3.2.2 Preprocessing of slides

The generated question should evaluate whether the student has understood the point that the teacher focused on. We believe that the teacher's focus can be obtained from the teaching slides. Therefore, this teaching content can be used in question generation. We store the text content of the slides in the slide set $Class_Page$ according to the page number and denote $Page_k$ as the text content of slide page k , where $Class_Page = \{Page_1, Page_2, \dots, Page_k, \dots\}$. The other function of the text in slides is filtering, which can be used to select the key content of speech. This process may refer to the content of the slideshow to use the sentences on

each page for filtering, so we store the sentence contents of each slideshow as $Page_k = \{Page_{Sentence_{k1}}, Page_{Sentence_{k2}}, \dots, Page_{Sentence_{kh}}, \dots\}$, where $Page_{Sentence_{kh}}$ represents the h^{th} sentence of slide page k .

3.2.3 Voice filter module

The text of $Page_Sentence_{km}$ of slides $Au_Sentence_{kh}$ will be sent to the pre-trained SBERT to obtain the sentence vector to be kept as $SBERT_{Page_Sentence_{km}}$ and $SBERT_{Au_Sentence_{kh}}$, and the cosine similarity of the two is calculated through the sentence vector. The formula is as follows.

$$sim_{Page_Sentence_{km}, Au_Sentence_{kh}} = \frac{SBERT_{Page_Sentence_{km}} \cdot SBERT_{Au_Sentence_{kh}}}{\|SBERT_{Page_Sentence_{km}}\| \times \|SBERT_{Au_Sentence_{km}}\|}$$

Finally, each sentence $Page_Sentence_{km}$ in the slideshow $Page_k$ on each page will be used to calculate the similarity value with $Au_Sentence_{kh}$ to determine the sentences with a phonetic similarity greater than α , and the phonetically similar sentences on each page will be collected. The set of similar sentences of page k is defined as $TPage_k = \{TPage_{k1}, TPage_{k2}, \dots, TPage_{kf}\}$, where slide page k is calculated to have f similar sentences that are put into $TPage_k$.

3.3 Answer generation module

Unlike past research that used an encoder-decoder neural network to obtain more semantic information, the proposed method uses a pretrained T5 model in answer generation. Although T5 is also a Transformer-based neural network model similar to existing pretrained language models, it finds the optimal structure of the language model and then uses it as a target. Many experiments have been performed on functions, datasets, training time, model size, multitask learning, etc. (Raffel et al., 2020).

3.3.1 Answer module input format

Since the training set used in this paper may have multiple sentences with answers in the input sentences, it needs to be formatted. For each sentence with an answer, the HL token training approach of Chan and Fan (2019) is used. It highlights sentences with the [hl] token, where the input sentences are marked as $\hat{C} = \{[hl], c_1, c_2, \dots, c_s, [hl], \dots, c_l\}$. c_s in \hat{C} is the ending word of a sentence in the training set. $\{c_1, c_2, \dots, c_s\}$ is the sentence with the marked answer in the training set. c_l represents the end of a sentence in the training set of one word. For the target answer to be generated, [sep] needs to be added to mark the end of the answer in the sentence. The output answer will be put into the form $\hat{A} = \{a_1, \dots, a_p, [sep]\}$, where a_p represents the word at the end of the answer.

3.3.2 Answer generation

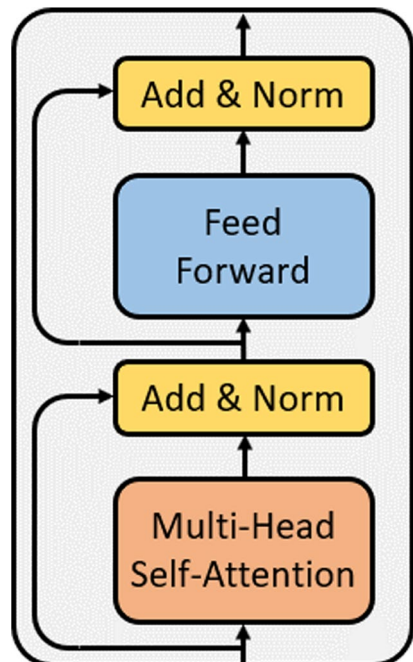
In this process, the input sentence $\hat{C} = \{[hl], c_1, c_2, \dots, c_s, [hl], \dots, c_l\}$ will go through 12 layers of encoder operations in total (Raffel et al., 2020). The sequence passes through the multihead self-attention mechanism layer, the first layer of the add & norm layer, the fully connected layer, and the second layer of the add & norm layer in turn, as shown in Fig. 3. The output of each layer is used as the encoder input of the next layer. The semantic expression of the target answer $\hat{A} = \{a_1, \dots, a_p, [sep]\}$ is output in the final decoder layer. Each element in the formula represents the semantics of an input token representation. The following will introduce the calculation method of each sub-layer in the single-layer encoder.

We define the input sequence as $InputSequence = \{a^1, a^2, \dots, a^i, \dots, a^n\}$. This input sequence can be compared to the abovementioned input sentence $\hat{C} = \{[hl], c_1, c_2, \dots, c_s, [hl], \dots, c_l\}$, where each item is a token that represents a word in the sentence. a^i represents the i^{th} token in this sequence, and n is the entire sequence length. First, a^i will be multiplied by three different matrices W^q , W^k , and W^v to generate three vectors q^i , k^i , and v^i . The formulas are as follows:

$$q^i = W^q a^i$$

$$k^i = W^k a^i$$

Fig. 3 Example of a Single-Layer Encoder



$$v^i = W^v a^i$$

The vector q^i stands for the query, and the main purpose is to match each key. The vector k^i represents the key that is to be matched by a query. The vector v^i is called the value and represents the information implicit in each token a^i . Next, we use the queries in the sequence to match each key with one of the queries. More specifically, the vectors q^i and k^i are calculated by scaled dot-product attention. Taking the input sequence $\{a^1, a^2, \dots, a^i\}$ as an example, the q^1 generated by a^1 is the corresponding $\{k^1, k^2, \dots, k^i\}$ of the input sequence. After multiplication, we have $\{\alpha_{1,1}, \alpha_{1,2}, \dots, \alpha_{1,i}\}$, where $\alpha_{1,i}$ is expressed as follows:

$$\alpha_{1,i} = q^1 \cdot k^i / \sqrt{d}$$

where the variable $\alpha_{1,i}$ is the attention weight of token q^1 and token k^i . The variable d represents the dimensions of q^1 and k^i . Then, $\alpha_{1,i}$ will be calculated through a softmax layer to generate $\hat{\alpha}_{1,i}$, which is used to control the proportion of information extracted from each token.

$$\hat{\alpha}_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,i})$$

Note that $\exp(\cdot)$ represents the exponential function, and the variable j represents the number of $\alpha_{1,i}$. Finally, we define the output sequence as $OutputSequence = \{b^1, b^2, \dots, b^i, \dots, b^n\}$. The token b^i represents the i -th output value in this sequence, and the variable n is the length of the entire sequence. Taking b^1 as an example, its value can be obtained by the following formula.

$$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$$

According to the above, for each token a^i in the input, a sequence will be calculated by the abovementioned formula to obtain each token b^i . Finally, the following formula represents the calculation of the attention mechanism.

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

where Q is a matrix that combines all tokens q^i , K is a matrix that combines all tokens k^i , V is a matrix that combines all tokens v^i , and the variable d_k represents the dimensions of Q and K . The multihead self-attention mechanism is used in T5. It generates multiple sets of q^i , k^i , and v^i according to each input a^i . The number of heads represents the number of sets generated, and the variable h represents the number of heads. Different heads are used to pay attention to different aspects. For example, some heads pay attention to local information, and some heads pay attention to global information. Compared with the self-attention mechanism, the multiple heads can be oriented toward more aspects of the text so that the generated context semantics can be more complete. Taking $h=2$ as an example, the original q^i , k^i , and v^i will each be split into two tokens; that is, q^i , k^i , and v^i will be split into $q^{i,1}$ and $q^{i,2}$, $k^{i,1}$ and $k^{i,2}$, and $v^{i,1}$ and $v^{i,2}$, respectively. Taking $q^{i,1}$ and $q^{i,2}$ as an example,

the formulas are as follows; analogous formulas are used for $k^{i,1}, k^{i,2}, v^{i,1}, v^{i,2}$ and so on, and $W^{q,1}$ and $W^{q,2}$ are two different matrices.

$$q^{i,1} = W^{q,1}q^i$$

$$q^{i,2} = W^{q,2}q^i$$

Then, $q^{i,1}, q^{i,2}, k^{i,1}, k^{i,2}, v^{i,1}, v^{i,2}$ follow. In the process mentioned above, $b^{i,1}, b^{i,2}$ are obtained through the calculation method of self-attention. The output b^i is calculated by the following formula.

$$b^i = \text{Concat}(q^{i,1}, q^{i,2})W^o$$

At this point, we can use the following formula to represent the calculation of the multihead self-attention mechanism layer.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_i, \dots, \text{head}_h)W^o$$

Here, $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$; the schematic diagram is shown in Fig. 4. In the method of this study, the multihead self-attention mechanism with $h=8$ is used as our model. Thus far, the calculation of the first sublayer is complete.

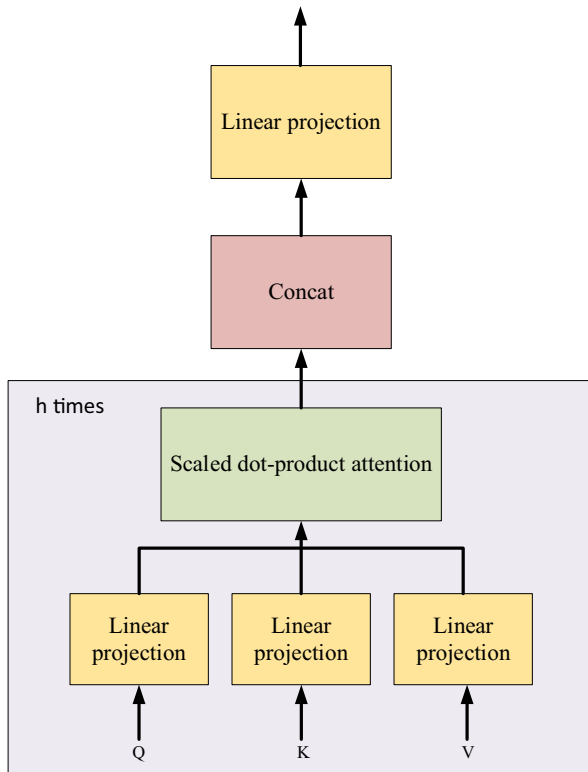


Fig. 4 Schematic Diagram of the Multihead Self-Attention Mechanism

The second sublayer is a positionwise fully connected feed-forward network. The fully connected layer contains two linear transformations, and an activation function is added between them. The formula is as follows.

$$FNN(x) = \sigma(xW_1 + b_1)W_2 + b_2$$

where x is $OutputSequence_{multi-head} = \{b^1, b^2, \dots, b^i, \dots, b^n\}$ and is obtained through the output of the add & norm layer, W_1 and W_2 are the weights that need to be learned for the network, b_1 and b_2 are bias parameters, and σ represents the activation function. Finally, the output of the fully connected layer will go through an add & norm layer to yield the output of the final single-layer encoder $T5_{SequenceAnswer} = \{d^1, d^2, \dots, d^n\}$.

3.3.3 Answer generation module

After 12 layers of encoders, the final hidden-layer vector is $T5_{SequenceAnswer}$, marked as Ha , and the target answer $\hat{A} = \{a_1, \dots, a_p, [sep]\}$ is generated in sequence from left to right. Possible answer words are detected. The architecture of the decoder is also composed of 12 layers of the same structure. Its internal architecture is similar to that of the encoder. It also includes two sublayers, a multihead self-attention mechanism, and a fully connected feedforward neural network, and each sublayer is added behind an add & norm layer. However, the self-attention in the decoder adds a masking mechanism to the encoder structure so that the decoder can prevent the self-attention mechanism from paying attention to the words after time point t when generating the words at the time point t position, ensuring that calculations at that point do not see future information.

In addition to this change, between the two sublayers of self-attention and the feedforward neural network, the decoder inserts a third sublayer, which is the multihead self-attention mechanism between the encoder and decoder and is different from the general self-attention mechanism. The difference between these attention mechanisms is that the query of encoder-decoder self-attention comes from the decoder, while the key and value come from the encoder. Through this design, the decoder can focus on the appropriate position in the encoder output according to different time points. A schematic diagram is shown in Fig. 5.

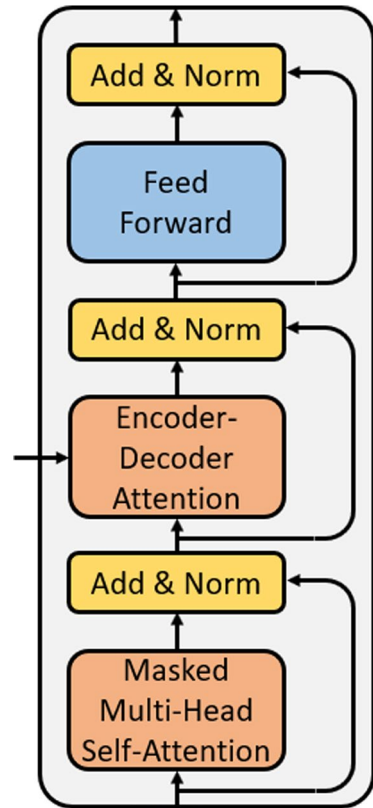
Finally, to predict the generated answer, the *Softmax* function is used to obtain the distribution probability of each word y_i in the generated answer. The input is the implicit vector Ha calculated by the decoder at the end, and it will be calculated until [sep] is generated. The formula is as follows.

$$p(y_i|y_0, \dots, y_{i-1}) = Softmax(W \cdot Ha)$$

3.3.4 Multiple-choice option generation module

In this study, to answer the questions of a distance course in real time, a Word2Vec similar word generation module is trained through the English Wiki corpus. The

Fig. 5 Example of a Single-Layer Decoder



word embedding used in this study is the Word2Vec method, which maps each word to a high dimension, finds its potential semantic representation, and encodes the relationship between words in the vector space. If the context of a word is similar to that of another word, then the two words are assumed to have similar meanings. Considering that Word2Vec has the advantages of saving computational space and employing simple concepts (Levy et al., 2015), the skip-gram model of Word2Vec will be used for training, and the reason for choosing skip-gram is that its algorithm aims to learn to predict a word context.

Therefore, this research uses the vocabulary set of all words in the English Wiki corpus $Word = \{w_1, w_2, \dots, w_t\}$ to pretrain with Word2Vec. After training, a new word embedding model that covers the pretrained vocabulary is obtained, $Vector = \{v_1, v_2, \dots, v_v\}$, where more similar words have more similar word vectors. Each variable v_v in $Vector$ is a d -dimensional vector representation of the v -th word in the word embedding model. Finally, according to the target answer A generated in the previous step, the model calculates the similarity through word vectors and queries the four most similar words to provide options for multiple-choice questions.

3.4 Question generation module

To generate questions, we fine-tune another pretrained T5 model. In the T5 network architecture, the goal of the model is to combine $\{c_1, c_2, \dots, c_s\}$ in the sentence \widehat{C} [hl] of the previous step with \widehat{A} . We take the answer $A = \{a_1, \dots, a_p\}$ of [sep], input it into the T5 network architecture, and generate a question $Q = \{q_1, q_2, \dots, q_u\}$ centered on the generated answer A , where q_u is the last word of the expected question and A is the target answer obtained in the previous stage. We combine answer generation and T5 pretraining to generate questions.

3.4.1 Question generation input sequence composition

In the same part of the input sequence, we refer to (Chan & Fan, 2019) to use the [hl] tag for training, then use [hl] in the input sentence to mark the position of the answer in the sentence, and use the [hl] tag to mark the generated answer A . The token is in the sentence, so the sentence becomes $C' = \{c_1, c_2, \dots, c_i, [hl], a_1, \dots, a_p, [hl], \dots, c_s\}$, where c_i is the representative word before the answer. For the question to be generated, we add [sep] to mark the end of the question in the sentence, and the output question becomes $Q' = \{q_1, \dots, q_u, [sep]\}$.

3.4.2 Question pretraining module

First, we define the input sequence $C' = \{c_1, c_2, \dots, c_i, [hl], a_1, \dots, a_p, [hl], \dots, c_s\}$; as with the answer generation module, a 12-layer encoder architecture is established with a T5 base. The task is to maximize the predicted output $Q' = \{q_1, \dots, q_u, [sep]\}$, and the input sequence of the multihead self-attention mechanism layer in the first layer becomes $InputQuestion_{multi-head} = \{g^1, g^2, \dots, g^n\}$. After the calculation ends at the multihead self-attention mechanism layer, the output will change to $OutputQuestion_{multi-head} = \{e^1, e^2, \dots, e^n\}$, representing the self-expression of the input sequence C' attention mechanism layer output. The second sublayer is a positionwise fully connected feed-forward network. The fully connected layer contains two linear transformations, and an activation function is added between them. The formula is as follows.

$$FNN(x) = (xW_1 + b_1)W_2 + b_2$$

where x is $OutputQuestion_{multi-head} = \{e^1, e^2, \dots, e^n\}$ and is obtained through the output of the add & norm layer, W_1 and W_2 are the weights that need to be learned for the network, b_1 and b_2 are bias parameters, and σ represents the activation function. Finally, the output of the fully connected layer will go through an add & norm layer to yield the output of the final single-layer encoder $T5_{SequenceQuestion} = \{d^1, d^2, \dots, d^n\}$.

3.4.3 Question generation module

The decoder of T5 marks the hidden layer vector output by the 12-layer encoder as Hq and generates the question $Q' = \{q_1, \dots, q_u, [sep]\}$ from left to right. This

task can be regarded as the detection of the encoder-generated question sentences. Finally, to predict the generated sentence, the function *Softmax* is used to obtain the distribution probability of each word y_i in the generated question sentence. The formula is as follows.

$$p(y_i|y_0, \dots, y_{i-1}) = \text{Softmax}(W \cdot Hq)$$

4 Results

4.1 Dataset

This study uses self-collected data, including video and slides from the graduate course Information Security, as evaluation data. To generate multiple-choice question options, this research uses the English corpus provided by Wikipedia for training, which contains the current Wikipedia titles, content, picture descriptions, and other text information, and uses the trained model to extract similar words from it. In addition, since a large amount of question–answer pair data is required for fine-tuning the pretrained model, it is insufficient to train only on the question sentences in the long-distance course, so this study uses the Stanford Question Answering Dataset (SQuAD) for model training. SQuAD was established by Rajpurkar et al. (2016); more than 100,000 question sentences were extracted from Wikipedia, and the construction of the questions and answers was mainly performed through crowd-sourcing, allowing annotators to propose up to 5 questions based on the content of the article with correct answers, where the answers had to appear in the original text. Although the content of the SQuAD dataset is not directly related to the course content, Wang et al. (2018) mentioned that training on a large dataset can effectively allow a model to learn how to ask and answer questions, which can still indirectly improve the question fluency degree and the correlation between the question and the original direct statement when the question is generated.

4.2 Evaluation measurement

In the sentence selection stage of speech filtering, since the task does not have a certain standard answer, this study refers to the evaluation method of Cheng and Lapata (2016) to manually evaluate the results of the system sentence selection. The evaluation benchmark refers to the standard of Agarwal and Mannem (2011) that the question–problem sentences in educational texts should have two criteria, Informativeness and Askability, to measure the quality of the sentence selection results. The evaluation of natural language generation systems can be divided into the quantity level and quality level. The quantity level is a numerical evaluation index. This study uses the bilingual evaluation understudy (BLEU) and recall-oriented understudy for gisting evaluation (ROUGE) based on n-gram matching. BLEU uses the measurement method of precision to calculate the n-gram similarity, while ROUGE is based

on recall and F-measure as the measurement methods. The quality level scores the sentences generated by this system through manual evaluation.

BLEU is a metric for evaluating generated sentences. The core idea is to compare the degree of overlap between the generated sentence and the n-gram in the reference sentence, where the n-gram refers to a segment composed of n consecutive words in a sentence. The value of the n-gram is usually 1 to 4, which indicates BLEU-1, BLEU-2, BLEU-3, or BLEU-4. The formula is as follows:

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

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')}$$

where n represents the n-gram, w_n represents the weight of the n-gram, and p_n is the accuracy score of the n-gram between two sentences. Because BLEU is based on the measurement method of accuracy, when the sentence length is shorter, the degree of repetition will be shorter as well. Long sentences obtain higher scores. To avoid bias in the score, BLEU introduces a length penalty factor (brevity penalty) in the final score results.

ROUGE is a metric for evaluating automatic summaries as well as translations; it measures the similarity between the automatically generated summaries or translations and reference summaries. ROUGE-N and ROUGE-L are often used in related research, where N in ROUGE-N refers to n-grams and L in ROUGE-L refers to the longest common subsequence (LSC). Nema and Khapra (2018), in their research on the evaluation index of question generation systems, mentioned that most research on question generation uses ROUGE-L as the evaluation index.

In ROUGE-L, the longest common subsequence is found between the sentence generated by the model and the reference sentence. This method assumes that the longer the longest common subsequence between two sentences is, the more similar the two sentences are. The difference between this method and the n-gram method is that the longest common subsequence does not need to be continuous. The formula is as follows.

$$Fscore = \frac{(1 + \beta^2)RP}{R + \beta^2P}$$

$$Precision = \frac{LCS(X, Y)}{n}$$

$$Recall = \frac{LCS(X, Y)}{m}$$

where $LCS(X, Y)$ represents the length of the longest common subsequence between X (a sentence generated by the model) and Y (a reference sentence), and n and m represent the lengths of the sentence generated by the model and the reference sentence, respectively. In the manual evaluation step, this study uses three manual

evaluation methods to verify the proposed model. The evaluation of multiple-choice options refers to the relatedness index proposed by Schnabel et al. (2015) to evaluate the relationship between words. This indicator scores the multiple-choice question options generated by the system. According to the correlation between the four options generated by the model and the original answer words, a score is assigned between 1 and 7, where 7 represents the highest correlation of the words.

To evaluate the question generation quality, the measurement method of Chen et al. (2019) evaluates the fluency of the generated question and the clarity relative to the original direct statement. For this evaluation method, enrolled students were invited to score the questions generated by the system according to the above two evaluation indicators by the following standards. (1) Fluency: Read the generated questions and assign 1–7 points for the fluency of the sentences and the correctness of the grammar. (2) Clarity: Compare the original direct statement and the generated question to decide whether the meaning remains after sentence conversion and whether the speech content of the course is related to the questions; assign 1–7 points as the score.

To evaluate the relationship between the answers and questions, as well as between the questions and courses, we refer to the relative indicators proposed by Huang et al. (2014) when evaluating the generated questions and the original text. The enrolled students graded the questions generated by the system according to the following indicators. (1) Answer-question correlation: For the generated answer, assign a score of 1–7 according to how closely it is related to the generated question. The higher the correlation is, the higher the score. (2) Question-course relevance: After reading the generated question, assign a score of 1–7 according to how relevant it is to the course content. The higher the relevance is, the higher the score.

4.3 Parameter setting

The question generation module in the final stage of this study uses T5 as the network architecture of the Transformer. In the parameter settings and optimization of T5, Raffel et al. (2020) used Adam as the training optimizer, the number of training epochs was set to 10, and the T5-base was used, with 12 hidden layers and 8 attention heads. Table 1 below shows the detailed network parameter settings:

Table 1 Parameter Settings of the Question Generation Model

Parameter	Value
Epoch	10
Batch size	32
Optimizer	Adam
Learning rate	0.00001
Embedding size	512
Vocabulary size	32,128
Number of blocks	12
Number of attention heads	8

4.4 Experiments

4.4.1 Experiment 1

Experiment 1 uses the speech similarity filtering method SBERT based on the pre-training model in the speech filtering module. It is expected to address the shortcomings of previous research methods that do not consider the semantic relationship of similarity. Therefore, the purpose of experiment 1 is to verify SBERT and sentences commonly used in previous research. Compared with the similarity filtering method, it is determined whether it can effectively select sentences that match the focus of the course. For the fairness of the experiment, the sentences whose similarity values are uniformly set to be greater than 0.5 will be removed, and the comparison methods are the pretrained models BERT and TF-IDF. (1) BERT: Using the BERT model pretrained on unlabeled data of different tasks and without using additional datasets for fine-tuning, the sentence vectors generated by BERT are used to compare the sentence similarity. (2) TF-IDF: The importance of words to documents is evaluated through statistical features, where TF is the word frequency and IDF is the inverse text frequency to deal with the problem of common words. The experimental method refers to the evaluation method of Cheng and Lapata (2016). Eight students who take the course are asked to read the class content and the audio file and compare the sentences filtered out in the module. The “Informativeness” measure is used to judge the relevance of the sentence as a whole to the original text and to judge whether too much noise has been included. The “Askability” score aims to judge how many of the filtered sentences could be generated as questions and to judge whether there are too many sentences that cannot be asked; points on a scale of 1–7 are given accordingly. These two indicators determine which sentence selection method selects the most suitable sentence for the question. It can be seen from the experimental results in Fig. 6 that SBERT has a score of more than 5 in terms of informativeness and askability from the perspective of classmates.

4.4.2 Experiment 2

Experiment 2 selects the speech filtering similarity parameters. It is obtained from the results of experiment 1. SBERT is the best similarity filtering model. The similarity score measures how well a method can filter out the sentences most relevant to the course content. In the numerical verification selection, 0.5 is the center, and a certain value is added or subtracted. In the test, it is found that if a sentence scores above 0.7, it cannot be filtered out, and sentences below 0.3 will have too much noise, so the similarity values between 0.3 and 0.7 are selected for manual evaluation. The experimental method is the same as the evaluation method of Cheng and Lapata (2016). Eight students who take the course are asked to read the class content and the audio file and compare the sentences filtered by the module. According to the informativeness of the sentences, the criterion of askability is scored on a scale of 1–7. From the experimental results in Fig. 7, it can be seen that when the parameter is set to 0.5, SBERT obtains higher scores in terms of informativeness

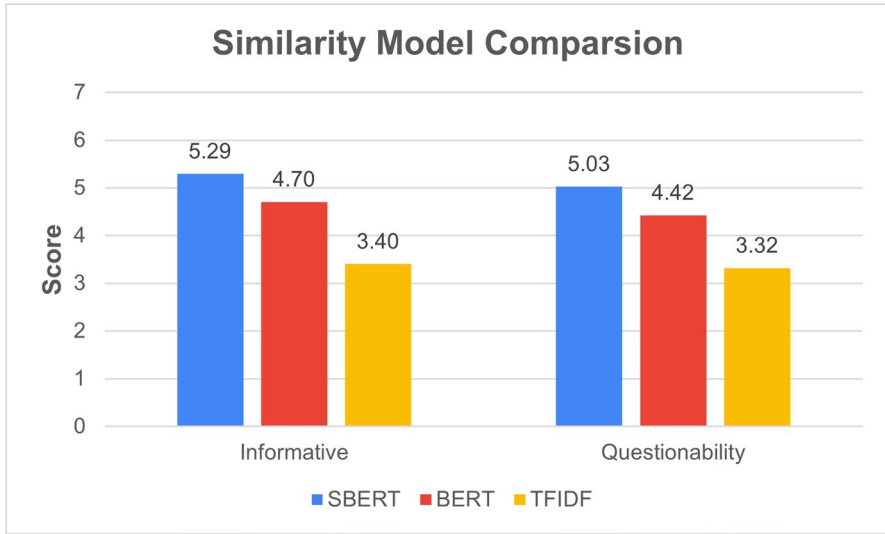


Fig. 6 Manual Evaluation of the Results of Speech Filtering Methods

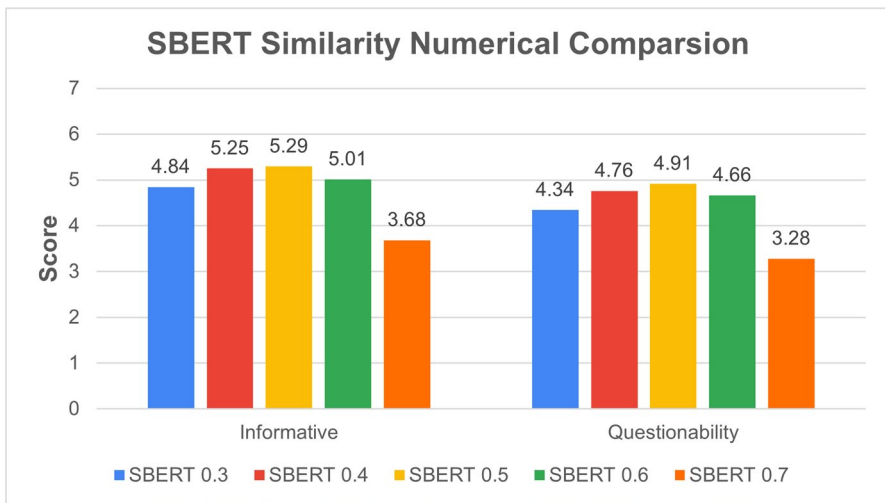


Fig. 7 Speech Filtering Similarity Parameter Evaluation Results

and askability. That is, when the parameter is set to 0.5, the results accord best with the perspective of the students.

4.4.3 Experiment 3

Experiment 3 manually evaluates the quality of the generated multiple-choice questions. This experiment evaluates whether the machine-selected option words are

appropriate, and the quality of the options is checked by human evaluation. The experimental method refers to the correlation index proposed by Schnabel et al. (2015) to evaluate the relationship between words. The evaluator will give the answer word generated by the model and the multiple-choice option generated by the model according to the correlation index between them. Each option is rated from 1–7. From the experimental results in Fig. 8, it can be seen that the first four generated options have correlation scores of more than 5. The similar word generation model trained by Word2Vec has certain effectiveness in option generation and access to highly relevant multiple-choice options.

4.4.4 Experiment 4

Experiment 4 aims to evaluate the effect of adding feature tags to the question generation model in this study. The BLEU and ROUGE-L indicators are used to evaluate the effect of adding [hl] feature tags to the question generation model. This experiment compares the following four methods: (1) the T5-Small (no [hl]) model without feature markers, (2) the T5-Small model with feature markers, (3) the asynchronous distance teaching-question generation (ADT-QG) (no [hl]) model without feature markers, and (4) the ADT-QG model with feature markers added. The experimental results are shown in Fig. 9. The ADT-QG model with feature markers and the T5 base achieves better performance in various indicators. It is confirmed that adding feature markers can enhance the generation effect.

4.4.5 Experiment 5

This experiment compares the performance of the ADT-QG model and the other answer-question generation methods on various automatic evaluation indicators.

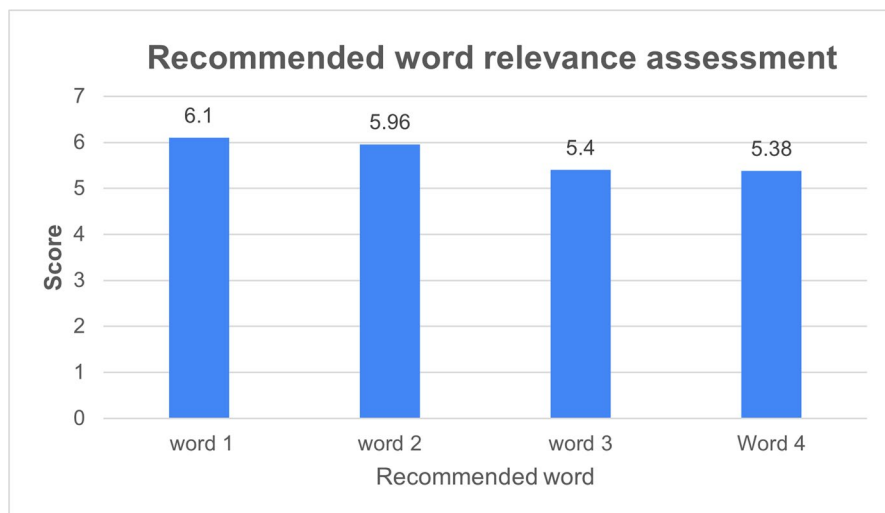


Fig. 8 Option Relevance Evaluation Results

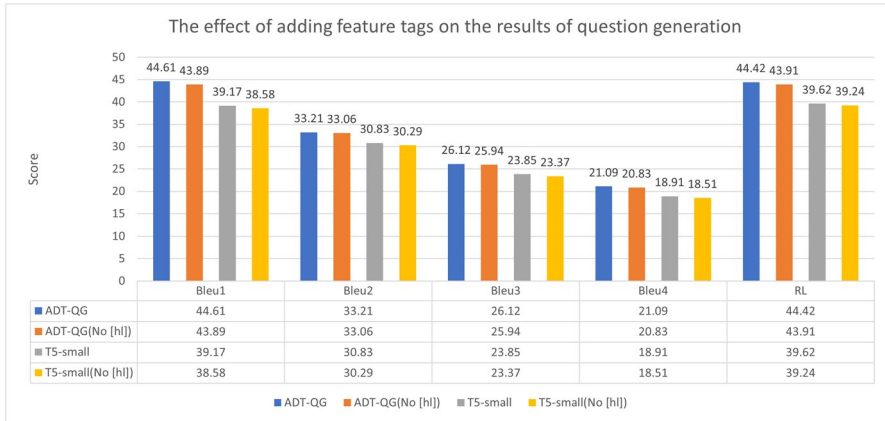


Fig. 9 The Effect of Adding Feature Tags on the Question Generation Results

(1) H&S: A rule-based AQG model proposed by Heilman and Smith (2010), which converts all sentences in the text into questions through artificially constructed syntactic rules and uses an overgenerate-and-rank approach. The policy selects the final output question. (2) L2A: Proposed by Du et al. (2017), the model consists of an encoder-decoder and an attention mechanism based on a recurrent neural network and uses long short-term memory (LSTM) as the network unit. (3) ASGEN: Kedia et al. (2019) proposed a two-stage question generation model for answer-question generation. BERT was used to extract the answer, and the Transformer was used to train the question generation model. Table 2 shows that through training the T5 Transformer and adding HL feature tags, the model can obtain local and global information, which can indeed improve the performance of the model in question generation.

4.4.6 Experiment 6

In Experiment 6, the question sentences selected from the course materials in the speech filtering module are used as input for the ADT-QG model and other question generation models. The quality of the questions is manually evaluated. The evaluator assigns 1–7 points for the fluency and clarity of the questions generated by the models. Since this method is mainly used for the evaluation of questions, the answers generated by ADT-QG and the original direct sentences are used to

Table 2 Automatic Evaluation of Question Generation Models

Model	BLEU_1	BLEU_2	BLEU_3	BLEU_4	ROUGE-L
H&S	28.77	17.81	12.64	9.47	31.68
L2A	38.06	22.76	15.72	11.46	37.84
ASGEN	43.26	31.12	24.31	19.41	42.89
ADT-QG	44.61	33.21	26.12	21.09	44.42

generate questions for the two models. Regarding fluency, only the questions generated by the two models are given, and the evaluator is asked to rate the fluency of the question after reading. For the clarity evaluation, the evaluator is asked to read the generated question and the original direct statement. Afterward, the students are asked whether the original meaning after the sentence pattern conversion was clear and whether the semantics were relevant.

The experimental results are shown in Fig. 10. The ADT-QG model is superior to the ASGEN model in all indicators, and the clarity related to learning is much higher. According to the evaluation results of the ADT-QG model alone, it has a performance of more than 5 points in both indicators. It can be seen that the questions generated by the T5 pretraining model and the original direct sentences are more fluent and clear and are better than those generated by the un-pretrained Transformer model in terms of semantic expression.

4.4.7 Experiment 7

This experiment evaluates the correlation between the answers generated by the ADT-QG model and the questions, as well as the questions and the curriculum, and examines the relevance of the questions using an artificial evaluation. The evaluator judges the relevance of the answers and questions and the questions and course based on the results generated by the two models, giving a score of 1–7. The experimental results are shown in Fig. 11. The ADT-QG model is superior to the ASGEN model in all indicators. It not only has better performance in terms of the correlation between the answers and the questions but also has a higher score for the correlation between the questions and the course. Looking at the evaluation results of the ADT-QG model alone, the performance of the two indicators reaches more than

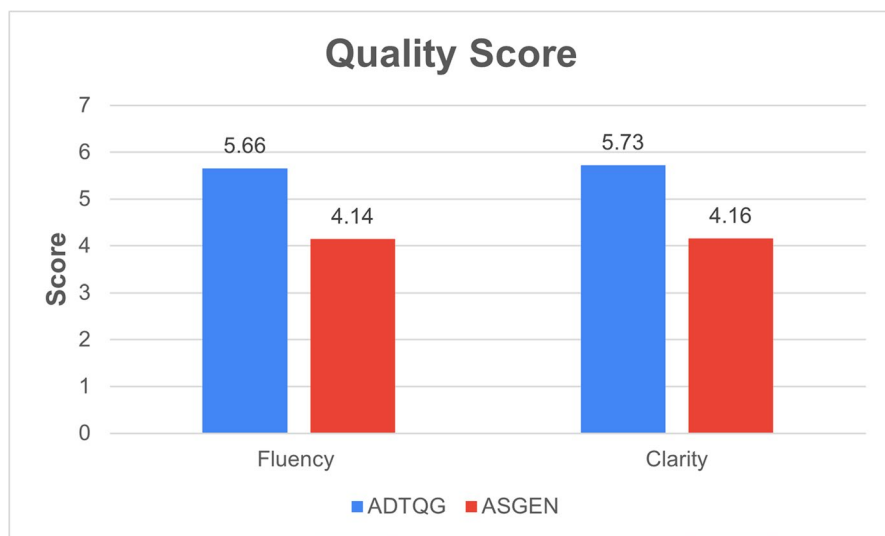


Fig. 10 Manual Evaluation Results of Question Generation

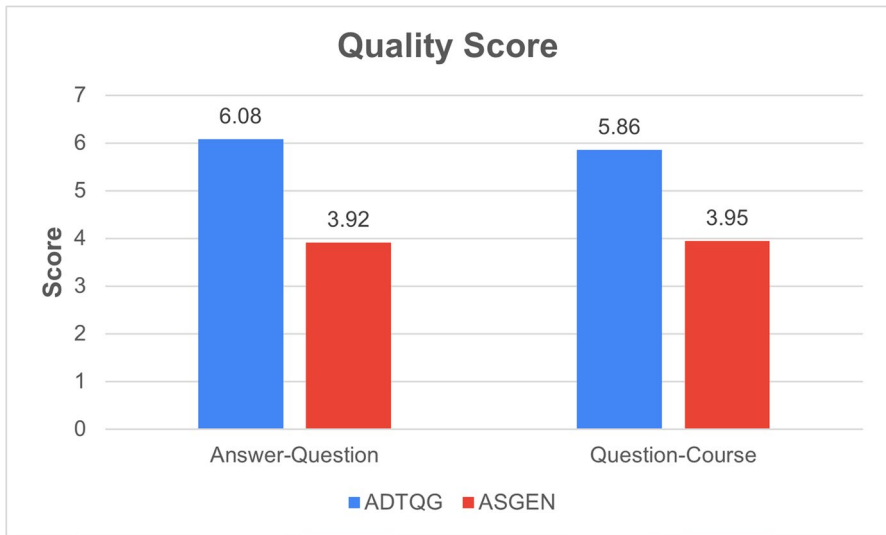


Fig. 11 The Results of Manual Evaluation of Question Relevance

5.5 points, and the score of the correlation between the answers and the questions reaches more than 6 points. It can be seen that the questions generated by the T5 pretraining model and the extracted answers are more in line with the key factors of the question required by the answer word. Additionally, the T5 pretraining model has good performance in the question-course correlation, so it can effectively generate questions related to the course. The answers and questions are more in line with the needs of the course, and the results are better than those of the Transformer model without pretraining.

5 Discussion

In this section, we will discuss the findings of all the experiments conducted in the preceding section. In experiment 1, we compared the SBERT approach to the BERT and TF-IDF approaches, which have been used in prior research to efficiently select sentences that correspond to the topics in a course. The speech filtering module uses multisource teaching materials and the sentence similarity filtering methods of SBERT, BERT, and TF-IDF to choose sentences that address the course's important concepts. In a prior study by Westermann et al. (2020), BERT was used to score the similarity of paragraphs. In this paper, the BERT model was evaluated using validation data extracted from existing datasets. This is identical to the study conducted by Lahitani et al. (2016), who examined the similarity between essay evaluations. In this study, the TF-IDF approach was used for the research model. In research employing the BERT and TF-IDF approaches, the concept of semantic relationships was not examined, providing the foundation for research undertaken to apply semantic relationships in the conducted trials. To evaluate the results of each technique, we

employed the manual evaluation method, which consists of surveying the course's enrolled students. In this study, we employed two criteria: informativeness and questionability. The findings of the experiment indicate that the content of the sentences selected using SBERT is superior to that of the sentences selected using the previous similarity filtering method. The highest manual evaluation score for informative indicators, with a value of 5.29, is obtained using the SBERT method, while a value of 4.70 is obtained using the BERT method, and a value of 3.40 is obtained using the TF-IDF method. The highest score for the askability or questionability indicator, with a score of 5.03, is obtained using the SBERT approach, while scores of 4.42 and 3.32, are obtained for the BERT and TF-IDF approaches, respectively.

Moreover, to capture a more accurate similarity value setting, the similarity setting for SBERT was also evaluated. The evaluation findings indicate that sentences can be filtered effectively when the value is set to 0.5. The highest manual evaluation scores, with values of 5.29 and 4.91, are obtained for the informative and askability indicators, respectively. Figure 7 shows that if the filter value is set to 0.7, the lowest scores for the informative and askability indicators, 3.68 and 3.28, respectively, are obtained using manual evaluation. The value chosen as the threshold will vary between studies. In a study conducted by Gaglani et al. (2020), a threshold value of 0.5 was determined, which is the same as the threshold determined in this investigation. Some researchers employed a cutoff value of 0.7 to define high coverage (Westermann et al., 2020). In grammatical error correction studies, 0.87 was established as the cutoff point (Didenko & Shaptala, 2019).

We analyzed the created multiple-choice questions in the next experiment. The indicators evaluated include the appropriateness of the word choice and the quality of the multiple-choice options. There are numerous ways to evaluate this experiment. Several assessment approaches can be employed to evaluate multiple-choice answers, according to prior research. The point-biserial correlation coefficient measures the link between a dichotomous item score and a continuous overall test score. The stronger the association between the item performance and total test score performance is, the more positive the correlation (Lai et al., 2016). In another study, Rodriguez-Torrealba et al. (2022) employed cosine similarity to determine the similarity of system-generated answer alternatives. Karmascore is an additional way to evaluate generated answer alternatives (Söbke, 2022). In this study, the authors sought human evaluation assistance. The model-generated answer and the model-generated multiple-choice options were evaluated based on the correlation index between them. Each choice was graded between 1 and 7. The first four response choices that were created had correlations greater than 5, with values of 6.1, 5.96, 5.4, and 5.38, respectively.

For the subsequent experiment (experiment 4), the model developed in this work, namely, the asynchronous distance teaching-question generation (ADT-QG) model, was evaluated in comparison to the T5-small model. In addition to comparing these two models, the ADT-QG and T5 approaches were compared with and without feature tags. To evaluate the models based on question creation, the BLEU and ROUGE-L indicators were employed. For BLEU evaluation, the BLEU1, BLEU2, BLEU3, and BLEU4 principles were utilized. The experiments indicate that the highest value for the ADT-QG model employing feature tags

(44.61) was obtained using BLEU1. The ADT-QG model without feature tags has the second-highest score (43.89), followed by the T5-small model with feature tags (39.17) and the ADT-QG model without feature tags (38.58). Experiments employing BLEU2, BLEU3, and BLEU4 evaluations all demonstrated the same pattern. ROUGE-L follows the same trend, where ADT-QG with feature tags has the highest value (44.42), followed by ADT-QG without feature tags (43.91), T5-small with feature tags (39.62), and T5-small without feature tags (39.24). It has been confirmed that the addition of feature markers may increase the generation effect. The results of experiment 4 are consistent with previous research. Several researchers have utilized feature tags to enhance the performance of question-and-answer generation algorithms (Kumari et al., 2022; Rao et al., 2022).

Through the automatic evaluation index and the SQuAD dataset, we compared the performance of the ADT-QG model in the fifth experiment. A comparison was made between the constructed model (ADT-QG) and models from previous research. The H&S model (Heilman and Smith, 2010), the L2A model (Du et al., 2017), and the ASGEN model (Kedia et al., 2019) were compared to the ADT-QG model. The results of the experiments show that the model proposed in this study has the best performance across all evaluation indicators: BLEU1 (44.61), BLEU2 (33.21), BLEU3 (26.12), BLEU4 (21.09), and ROUGE-L (44.42). The lowest score across all evaluation indicators is obtained using the H&S model. The L2A model is the third best model, and the ASGEN model is the second best model after the model proposed in this study. The sixth experiment was a continuation of the preceding experiment. This experiment involved a manual examination of the question phrases generated by the model proposed in this study. The model in this work (ADT-QG) was compared with that by Kedia et al. (2019) (ASGEN). Fluency and clarity served as the indicators for this manual evaluation. According to the testing data, the ADT-QG model performs better than the ASGEN model for both indices. Scores of 5.66 out of 7 for the fluency indicator and 5.73 out of 7 for the clarity indicator are obtained using ADT-QG. Scores of 4.14 for fluency and 4.16 for clarity are obtained using ASGEN.

The final experiment involved a comparison between the model proposed in this work (ADT-QG) and the ASGEN model (Kedia et al., 2019). In this study, we examined the correlation between questions and answers generated by the ADT-QG and ASGEN models. The correlation between questions generated by the present model and curriculum courses was also assessed. It can be observed from these two assessment types that our model (ADT-QG) achieves the highest correlation value based on manual evaluation. The score of 6.08 reflects the correlation between the responses to the ADT-QG questions, whereas the score of 3.92 reflects that of the ASGEN model. A score of 5.86 is achieved by ADT-QG for question-course correlation, while a score of 3.95 is obtained for ASGEN. The conclusion of this experiment is that in the evaluation of multiple-choice option generation, the options generated by the method proposed in this study have a high correlation with the original answers and can aid students in responding to questions from a distance learning course in a timely manner. Then, questions from a teacher's speech were filtered in the speech filtering module. The question generating outcomes from these questions as well as the long-distance

course questions generated by the ADT-QG model and the ASGEN model were evaluated artificially.

6 Conclusion and future work

With the advancement of technology and the diversified development of learning methods, teaching is no longer limited to classrooms. Due to the impact of COVID-19, distance teaching has become popular. However, the teacher may find that students cannot pay attention to the class. Therefore, this study aims to use technology to help teachers attract students' attention through automatically generated questions. Existing question generation methods mostly use artificially designed rules or templates to generate questions, but this is labor intensive and time consuming. Therefore, in recent years, research has begun to adapt pretrained language models for question generation. However, most research focuses on the generation performance of questions and not the quality of the questions. Therefore, this study proposes an asynchronous distance-teaching question generation method to generate quiz questions. To confirm whether students are paying attention to the lectures, this study uses the teacher's statements and the content of the textbook to generate multiple-choice questions. To achieve this goal, the proposed method extracts keywords from the teacher's lecture and compares the electronic textbook to select the important sentences as the source of the questions. By pretraining the language model and adding feature tags, the different semantics of words in different sentences can be considered when judging the answer and constructing the question. An option generation module is also added to increase the immediacy of answering questions in distance teaching through multiple-choice questions.

In this research, seven different experiments are conducted. Experiment 1 employs the SBERT approach for speech similarity filtering based on the speech filtering module in the pretraining model. It is expected to address the shortcomings of prior research approaches that disregard the semantic relationship of similarity. Experiment 2 is performed to determine the parameters for speech filtering similarity. Experiment 3 is a manual evaluation of the quality of generated multiple-choice questions. In this experiment, the appropriateness of the machine-selected option phrases is evaluated, and the quality of the options is assessed manually. The objective of experiment 4 is to assess the impact of adding feature tags to the question generating model in this study. The BLEU and ROUGE-L indicators are utilized to assess the impact of adding [hl] feature tags to the question generation model. In experiment 5, the performance of the ADT-QG model is compared with that of other answer-question generating methods using a variety of automatic evaluation markers. In experiment 6, question phrases taken from course materials in the voice filtering module are input into the ADT-QG model and other question generating models. The quality of questions is evaluated manually. Experiment 7 explores the link between the responses provided by the ADT-QG model and the questions, as well as the questions and the curriculum, and the relevancy of the questions using a fake evaluation. The results show that the questions generated by the ADT-QG model proposed in this study

achieve better performance in the fluency and clarity indicators, indicating that the questions generated by the ADT-QG model have a certain quality and are relevant to the curriculum. The method also achieved good performance in the evaluation, indicating that the questions generated by the ADT-QG model are highly related to the curriculum, so the questions generated in this study have the potential to help teachers automate questions or help students enhance their attention in the classroom.

Funding The research is based on work supported by Taiwan Ministry of Science and Technology under Grant No. MOST 107–2410-H-006 040-MY3 and MOST 108–2511-H-006–009. We would like to thank partially research grant supported by "Higher Education SPROUT Project" and "Center for Innovative FinTech Business Models" of National Cheng Kung University (NCKU), sponsored by the Ministry of Education, Taiwan.

Data availability Data sharing does not apply to this article, as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors of this study declare no conflicts of interest.

References

- Agarwal, M., & Mannem, P. (2011). *Automatic Gap-Fill Question Generation from Text Books*. Paper presented at the 6th Workshop on Innovative Use of NLP for Building Educational Applications, Portland, Oregon.
- Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32), 15849–15854.
- Belt, E. S., & Lowenthal, P. R. (2022). Synchronous video-based communication and online learning: an exploration of instructors' perceptions and experiences. *Education and Information Technologies*, 1–24. <https://doi.org/10.1007/s10639-022-11360-6>
- Blau, I., Weiser, O., and Eshet-Alkalai, Y. (2017). How do medium naturalness and personality traits shape academic achievement and perceived learning? An experimental study of face-to-face and synchronous e-learning. *Research in Learning Technology*, 25. <https://doi.org/10.25304/rlt.v25.1974>
- Chan, Y.-H., & Fan, Y.-C. (2019). *A recurrent BERT-based model for question generation*. Paper presented at the Proceedings of the 2nd Workshop on Machine Reading for Question Answering, Hong Kong, China.
- Chen, G., Yang, J., & Gasevic, D. (2019). *A Comparative Study on Question-Worthy Sentence Selection Strategies for Educational Question Generation*. Paper presented at the the International Conference on Artificial Intelligence in Education, Chicago, USA.
- Cheng, J., & Lapata, M. (Writers). (2016). Neural Summarization by Extracting Sentences and Words. In *the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany*
- Debes, G. (2021). Distance learning in higher education during the COVID-19 pandemic: Advantages and disadvantages. *International Journal of Curriculum and Instruction*, 13, 1109–1118.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding*. Paper presented at the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota.
- Dhawan, S. (2020). Online learning: A panacea in the time of COVID-19 crisis. *Journal of Educational Technology Systems*, 49(1), 5–22.
- Didenko, B., & Shaptala, J. (2019). *Multi-headed architecture based on BERT for grammatical errors correction*. Paper presented at the Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications, pp. 246–251.



- d'Inverno, R., Davis, H., & White, S. (2003). Using a personal response system for promoting student interaction. *Teaching Mathematics and Its Applications: International Journal of the IMA*, 22(4), 163–169.
- Du, X., Shao, J., & Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.
- Fabriz, S., Mendzheritskaya, J., & Stehle, S. (2021). Impact of Synchronous and Asynchronous Settings of Online Teaching and Learning in Higher Education on Students' Learning Experience During COVID-19. *Frontiers in Psychology*, 12, 733554. <https://doi.org/10.3389/fpsyg.2021.733554>
- Gaglani, J., Gandhi, Y., Gogate, S., & Halbe, A. (2020). *Unsupervised whatsapp fake news detection using semantic search*. Paper presented at the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).
- Gamage, K. A. A., Gamage, A., & Dehideniya, S. C. P. (2022). Online and Hybrid Teaching and Learning: Enhance Effective Student Engagement and Experience. *Education Sciences*, 12(10). <https://doi.org/10.3390/educsci12100651>
- Heilman, M., & Smith, N. A. (2010). *Good Question! Statistical Ranking for Question Generation*. Paper presented at the the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California.
- Hrastinski, S. (2008). Asynchronous and Synchronous e-Learning. *Educause Quarterly*, 31(4), 51–55.
- Huang, Y.-T., Tseng, Y.-M., Sun, Y. S., & Chen, M. C. (2014). *TEDQuiz: automatic quiz generation for TED talks video clips to assess listening comprehension*. Paper presented at the 2014 IEEE 14th international conference on advanced learning technologies. Athens, Greece
- Kaplan, A. M., & Haenlein, M. (2016). Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster. *Business Horizons*, 59(4), 441–450.
- Kedia, A., Chinthakindi, S. C., Back, S., Lee, H., & Choo, J. (2019). ASGen: Answer-containing Sentence Generation to Pre-Train Question Generator for Scale-up Data in Question Answering. In *International conference on learning representations*. Addis Ababa, Ethiopia
- Kumari, V., Keshari, S., Sharma, Y., & Goel, L. (2022). *Context-Based Question Answering System with Suggested Questions*. Paper presented at the 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence).
- Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). *Cosine similarity to determine similarity measure: Study case in online essay assessment*. Paper presented at the 2016 4th International Conference on Cyber and IT Service Management.
- Lai, H., Gierl, M. J., Touchie, C., Pugh, D., Boulais, A.-P., & De Champlain, A. (2016). Using automatic item generation to improve the quality of MCQ distractors. *Teaching and Learning in Medicine*, 28(2), 166–173.
- Lemay, D. J., Bazalais, P., & Doleck, T. (2021). Transition to online learning during the COVID-19 pandemic. *Computers in Human Behavior*, Rep, 4, 100130. <https://doi.org/10.1016/j.chbr.2021.100130>
- Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3, 211–225.
- Lin, K.-Y. (2015). Evaluating the effect of a clicker in an information literacy course for college nursing students in Taiwan. *CIN: Computers, Informatics, Nursing*, 33(3), 115–121.
- Liu, S., Li, Z., Zhang, Y., & Cheng, X. (2019). Introduction of key problems in long-distance learning and training. *Mobile Networks and Applications*, 24(1), 1–4.
- Martha, A. S. D., Junus, K., Santoso, H. B., & Suhartanto, H. (2021). Assessing undergraduate students' e-learning competencies: A case study of higher education context in Indonesia. *Education Sciences*, 11, 189. <https://doi.org/10.3390/educsci11040189>
- Masalimova, A. R., Khvatova, M. A., Chikileva, L. S., Zvyagintseva, E. P., Stepanova, V. V., & Melnik, M. V. (2022). Distance Learning in Higher Education During Covid-19. *Frontiers in Education*, 7. <https://doi.org/10.3389/feduc.2022.822958>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA*.
- Mitkov, R. (2003). *Computer-aided generation of multiple-choice tests*. Paper presented at the Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing.
- Mostow, J., & Jang, H. (2012). *Generating diagnostic multiple choice comprehension cloze questions*. Paper presented at the Proceedings of the Seventh Workshop on Building Educational Applications Using NLP.
- Nema, P., & Khapra, M. M. (2018). *Towards a Better Metric for Evaluating Question Generation Systems*. Paper presented at the the 2018 Conference on Empirical Methods in Natural Language Processing.
- Önoral, Ö., & Kurtulmus-Yilmaz, S. (2020). Influence of COVID-19 pandemic on dental education in cyprus: Preclinical and clinical implications with E-learning strategies. *Advanced Education*, 7, 69–77.

- Patra, R., & Saha, S. K. (2018). A hybrid approach for automatic generation of named entity distractors for multiple choice questions. *Education and Information Technologies*, 24(2), 973–993. <https://doi.org/10.1007/s10639-018-9814-3>
- Pennington, J., Socher, R., & Manning, C. (2014). *Glove: Global Vectors for Word Representation*. Paper presented at the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar.
- Perveen, A. (2016). Synchronous and asynchronous e-language learning: A case study of virtual university of Pakistan. *Open Praxis*, 8(1), 21–39.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67.
- Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *Squad: 100,000+ Questions for Machine Comprehension of Text*. Paper presented at the the Empirical Methods in Natural Language Processing (EMNLP), Austin, Texas, USA.
- Rao, P. R., Jhavar, T. N., Kachave, Y. A., & Hirlekar, V. (2022). *Generating QA from Rule-based Algorithms*. Paper presented at the 2022 International Conference on Electronics and Renewable Systems (ICEARS).
- Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Rodríguez-Torrealba, R., García-López, E., & García-Cabot, A. (2022). End-to-End generation of Multiple-Choice questions using Text-to-Text transfer Transformer models. *Expert Systems with Applications*, 208, 118258.
- Roh, H.-K., & Lee, K.-H. (2017). A Basic Performance Evaluation of the Speech Recognition APP of Standard Language and Dialect using Google, Naver, and Daum KAKAO APIs. *Asia-Pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, 7(12), 819–829.
- Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015). *Evaluation methods for unsupervised word embeddings*. Paper presented at the Proceedings of the 2015 conference on empirical methods in natural language processing. Lisbon, Portugal
- Söbke, H. (2022). Exploring (Collaborative) Generation and Exploitation of Multiple Choice Questions: Likes as Quality Proxy Metric. *Education Sciences*, 12(5), 297.
- Spector, J. M., Merrill, M. D., Elen, J., & Bishop, M. J. (2014). *Handbook of research on educational communications and technology*. Springer.
- Wang, Z., Lan, A. S., Nie, W., Waters, A. E., Grimaldi, P. J., & Baraniuk, R. G. (2018). *QG-Net: a Data-Driven Question Generation Model for Educational Content*. Paper presented at the the Fifth Annual ACM Conference on Learning at Scale, London, United Kingdom.
- Westermann, H., Savelka, J., & Benyekhlef, K. (2020). *Paragraph similarity scoring and fine-tuned BERT for legal information retrieval and entailment*. Paper presented at the JSAI International Symposium on Artificial Intelligence.
- Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-i., & Jegelka, S. (2020). How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*.
- Zagouras, C., Egarchou, D., Skiniotis, P., & Fountana, M. (2022). Face to face or blended learning? A case study: Teacher training in the pedagogical use of ICT. *Education and Information Technologies*, 27(9), 12939–12967. <https://doi.org/10.1007/s10639-022-11144-y>
- Zhang, K., & Wu, H. (2022). Synchronous Online Learning During COVID-19: Chinese University EFL Students' Perspectives. *SAGE Open*, 12(2). <https://doi.org/10.1177/21582440221094821>
- Zhou, W., et al. (2020). "Pre-training text-to-text transformers for concept-centric common sense." arXiv preprint [arXiv:2011.07956](https://arxiv.org/abs/2011.07956).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Hei-Chia Wang^{1,2}  · Martinus Maslim^{1,3}  · Chia-Hao Kan¹

Martinus Maslim
martinus.maslim@uajy.ac.id

Chia-Hao Kan
r76094153@mail.ncku.edu.tw

- ¹ Institute of Information Management, College of Management, National Cheng Kung University, Tainan City, Taiwan
- ² Center for Innovative FinTech Business Models, National Cheng Kung University, Tainan City, Taiwan
- ³ Informatics Department, Faculty of Industrial Technology, Universitas Atma Jaya Yogyakarta, Depok, Daerah Istimewa Yogyakarta, Indonesia