Comparison of template-based and multilayer perceptron-based approach for automatic question generation system

Walelign Tewabe Sewunetie[1,2], László Kovács[1]
[1]József Hatvany Doctoral School of Computer Science and Engineering, Faculty of Mechanical Engineering and Informatics, University of Miskolc, Miskolc, Hungary
[2]Department of Information Technology, School of Computing, Institute of Technology, Debre Markos University, Debre Markos, Ethiopia

Article Info

Article history:

Received Nov 3, 2021
Revised Aug 22, 2022
Accepted Sep 16, 2022

Keywords:

Automatic question generation
Intelligent tutoring system
Multilayer perceptron-based AQG
Phrase-level AQG
Template-based AQG

ABSTRACT

Intelligent tutoring systems are computer-assisted learning systems with adaption to students using artificial intelligence tools. An intelligent tutoring system can drastically improve education efficiency as it provides solutions to many issues that now plague the educational industry. One important component in education is questioning learners to assess and reinforce learning. This research compares two approaches for automatic question generation, a template-based question generation strategy and the phrase-Level automatic question generation system utilizing the Multilayer perceptron model. A template-based technique is a baseline for automatic question generation that uses templates taken from the training set to generate questions by filling certain templates with specific topic items. We utilize question-answer sentence composition datasets and manually constructed datasets for our experiments and comparison of the Multilayer perceptron training model. We used both human and automatic evaluation metrics to assess the efficiency of our suggested methods. Regarding automatic metrics, we selected the bilingual evaluation understudy (BLEU-n) gram and recall-oriented understudy for gisting evaluation (ROUGE-N) methods. The evaluation results demonstrate that the phrase-level multilayer perceptron-based strategy dominates the template-based approach and has a promising score in both ROUGE automatic and human evaluation metrics.

Corresponding Author:

Walelign Tewabe Sewunetie
József Hatvany Doctoral School of Computer Science and Engineering
Faculty of Mechanical Engineering and Informatics, University of Miskolc
Miskolc, Hungary
Email: walelignewabe@gmail.com

1. INTRODUCTION

Intelligent utoring systems (ITS) are computer programs that customize and enhance automation in teaching and online education [1], by utilizing artificial intelligence technology (AIT). The use of AIT to promote remote teaching and learning by integrating multiple resources, assisting teachers in the deployment of online classes, and assisting students in learning courses online, among other things, is referred to as online education. Because the learning environment is always changing, the learning platform has had to adapt and personalize its learning resources for students [2]. Online education has become an inevitable alternative and a mainstream type of education in a number of countries as a result of COVID-19. ITSs providing web-based high-quality curriculum resources for students to study at home using platform-based applications that enabling teachers to teach without face-to-face coaching, online education has evolved into a common aspect of education [3]. In different study work [2] a list of learner model attributes proposed in dynamic learning to support adaptation and personalization.

Self-assessment by students, according to self-regulated learning theories, can encourage in-depth reflection and assist driving effective self-regulated learning [4]. However, little research has been done on the relationship between students' self-evaluation and learning outcomes in ITSs [4]. One of the new approach which presented in [5] is a multi-agent-based e-learning system that is developed to support the assessment of prior learning skills in students. Under the banner of computer assisted instruction (CAI), the computer was initially introduced to the field of education in the 1970s. Carbonel outlined efforts to use computers in education in the 1970s. He argued that by adding artificial intelligence (AI) approaches to overcome current constraints, a CAI may be provided with greater capabilities [1].

In recent decades, the name ITS has frequently been substituted for intelligent computer assisted instruction (ICAI). In advanced learning technologies, such as ITS, game-based learning environments, and research-based environments [6], asking assessment questions are a key component. A common human strategy for producing questions is to read the article thoroughly, creating an internal model of knowledge, and then asking questions based on that model. The ability to automatically generate inquiries is one of the most significant barriers to the widespread adoption of ITS. Automatic question generation (AQG) has shown to be one of the most essential applications to solve this challenge, given that the act of questioning has been found to increase students' learning outcomes. The workload and reliance on humans will be reduced by automating the question creation process.

In the instance of AQG, the user enters text into the engine, which automatically generates a list of questions. From multiple input forms such as text, a structured database, or a knowledge base, AQG is defined as the challenge of creating syntactically sound, semantically valid, and acceptable questions [7]. Furthermore, the questions should be pertinent to the book and should be based on the text's replies. Various question generation approaches can provide a question that measures learners' knowledge in a variety of ways. Students and the system may not have direct interaction with the question generation system in the extended ITS design [8]. Our study focus on building the main part of the extended ITS design which is AQG model. Even though the AQG researchers began their work a decade ago, they still have serious shortcomings.

The main goal of our investigation was to compare two very different question generation techniques on a closed domain. The first version is a traditional template-based method which requires manual rule generation. Although the template-based solution is not an option for an open domain, we can generate appropriate ruleset for a closed limited domain. The second approach is the neural network-based method using now a multilayer perceptron (MLP) architecture.

In the following section, we present an overview of related works and common AQG techniques. In section 3 we discussed the proposed AQG method and its system architecture in more detail. In section 4 we explain our dataset organization, implementation, and test description. In section 5 we report on report on evaluation results compare our approach to existing question generation, before concluding in section 6. The main contribution of this study is to develop AQG model using a phrase based MLP and template-based approachs. In addition, the author test and analyze the efficiency of both approachs.

The paper presents an efficiency comparison of two AQG methods, the template-based and the neural network-based methods on a closed domain. The performed test experiences show that both methods can provide a good question set, but the neural network-based method using NLP dominates the classic template-based approach not only in the open world domain but also in the closed word domain.

2. RELATED WORK

The related works section, the literature should be presented in a way that connects the studies in a logical sequence. It is more sufficient for readers to read sequenced and connected paragraphs than reading separated paragraphs. Researchers from other disciplines have recently become interested in the research topic of AQG for educational objectives. Cohen [9] proposed that the substance of a question can be represented as an open formula with one or more unbound variables in one of the first works on questions. While question generation research has been done for a long time, the use of AQG for educational purposes has attracted the attention of several academic communities in recent years. Questions have also been a major topic of study in computational linguistics where models of the transformation from answers to questions have also been developed [10]. The research work from Rus and Lester [11] has directly approached the topic of generating questions for educational purposes. Mitkov *et al.* [12] demonstrated that automatic generation and manual correction of questions can be more time-efficient than manual authoring alone. Mitkov and Chen [13] created an automated reading tutor that uses AQG to help students improve their comprehension skills while reading a text. They looked at ways to automatically construct self-questioning instruction based on assertions in narrative texts about mental states (e.g., belief, intention, hypothesis, and emotion). This system employs a template-based approach to question generation.

The template-based strategy, according to the researchers [14], use templates taken from the training set to generate questions, which are subsequently filled with specific topic items. A study named "automatic chinese question generation (ACQG)" [15] employed a template-based method to create questions from key sentences and an adapted text rank to extract key sentences. After analyzing their findings, they come to the conclusion that due to the limited number of templates, the sorts of queries generated are equally limited. They advocate building enough text and question pairs in the future to train an end-to-end neural model to create high-quality Chinese questions directly [15]. Keklik *et al.* [16] in sentence-to-question transformation uses external rules or internal templates to modify the syntactic representation of the supplied input sentence. Syntax-based techniques use a standard procedure to decode a phrase to assess its syntactic structure, simplify it, if necessary, recognize significant phrases and apply syntactic transformation rules, and request word substitution [17]. They employed syntactic transformations for QG in Agarwal *et al.* [18], and the system has the question type, auxiliary, and content. An auxiliary verb is a verb that seems to communicate stress, aspect, modality, expression, emphasis, and other things to the clause in which it appears [19]. Jouault and Seta [20] proposed to construct semantics-based questions by accessing semantic information from the wikipedia database to improve learners' self-directed learning, in contrast to approaches that use text as an input. Previous research on question generation has mostly relied on hard heuristic principles to convert a sentence into questions [21], [22].

Serban *et al.* [23] suggested that a neural network method for generating factual questions from structured data instead of producing questions from texts. Zhou *et al.* [24] performs a preliminary investigation on question creation from text using neural networks, dubbed the neural question generation (NQG) framework, to produce natural language questions from text without the use of pre-defined criteria. The advanced natural language processing (NLP) techniques employed for the textual question creation include natural language understanding (NLU) and natural language generation (NLG) [25]. First, the system has to understand the input text which is NLU, and then it has to generate questions also in the form of text that is NLG. Blšták and Rozinajová [26] presented a system for generating factual inquiries from unstructured material. They combine numerous machine learning (ML) algorithms with classical linguistic methodologies based on sentence patterns. In the disciplines of NLP and computer vision, generating natural language queries for picture understanding is a hot topic [27]. Regarding the implementations of the learning modules the most dominant solution is neural network-based architecture specially the MLP and recurrent neural networks [28]. ML was primarily employed in the visual question generation (VQG) method to produce image captions. Using NLP algorithms, the image caption is converted into a question. VQG blends NLP, which allows the inquiry to be generated, with computer vision techniques, which allow the image's content to be understood [25].

The expected benefits of question generation from a given text using a neural network are: the training data should require little or no human effort and should reflect commonly-asked question intentions; the questions are generated based on natural language passages and should be of good quality, and the generated questions should be useful to QA tasks. According to previous research [28], neural-based AQG obtains large-scale, high-quality training data via the community-QA (CQA) website. The use of deep neural networks to extract target responses from a given article or paragraph and generate questions based on the target answers is known as neural question generation NQG [29].

In our tests, the proposed method reached an 85% accuracy level compared with the question generated by humans. This means an improvement to the results of [30] where a semi-automated method using NLP techniques to generate grammatical test items was developed based on the experiences, the method had generated 77% meaningful questions. A similar accuracy was achieved by [31] where the human-based evaluation revealed that their system produced 80% worthy cloze test items.

3. PROPOSED AQG METHOD

3.1. Template-based architecture

The notion behind template-based question generating is that a question template can capture a class of context-specific category questions. The block diagram of template based AQG process from sentence to question generation has been illustrated in Figure 1. The proposed system ruleset and matching dictionary has been done for filling templates.

We notice that the formulas which is not cited mentioned in this paper are formulated by authors. In the development of a template-based question generation system; rule set constructions phase is the core basic module. A template rule is a pair of sentence pattern and question pattern; *R [ST, QT]*, where the sentence pattern and the question pattern are given with a list of tokens:

$$\mathrm{ST} = [T^S_{1,} \, T^S_{2,} \dots T^S_m] \tag{1}$$

where $T_{i,}^{S}$ ∈ Tokens and Tokens denotes the set of tokens which are references for a part of speech (POS) tag and a position description value:

$$\mathrm{QT} = \{[T_{1,}^{Q}\ T_{2,}^{Q} \ldots T_{m}^{Q}]\} \tag{2}$$

where T_{i}^{Q} ∈ Words or $T_{i,}^{Q} \in ST$ symbol Words denotes the set of valid words of the language. In the question generation, the first step is to determine the token list T for the input sentence S:

$$T=[T_1, T_2, \ldots T_m], \\ T_i \in Tokens \tag{3}$$

Then the best matches rule is selected from the rule set to generate the question sentence.

$$R_w=\mathrm{argmax}_r\{\mathrm{sim}(T, ST_r)\} \tag{4}$$

The symbol sim () means a similarity calculation of sentence patterns. In our experiment we use the formula sim=$\frac{1}{1+edit\ distance\ (T,STr)}$ for similarity calculation of two strings. The idea behind the measure is that it would approximate a post-editor, in that it is based on edit distance, i.e., the minimum number of deletions, substitutions and insertions of words needed to turn the candidate translation into the reference translation [32]. The output question sentence is constructed with the application of R_w on S. Q=apply (R_w, S), where the apply method converts QT_w into a sequence of words.

$$Q = \begin{cases} T_i^Q \in QTw \ if \ T_i^Q \in \ Words \\ \ sub(T_i^Q) \ if \ T_i^Q \notin \ Words \end{cases} \tag{5}$$

The symbol $sub(T)$ denotes the function to determine the corresponding word for token T from S.

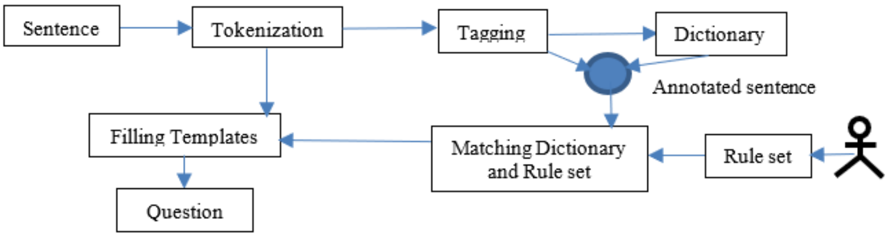


Figure 1. Block diagram of template-based AQG

The detailed implementation for the best match ruleset selection techniques is illustrated in Algorithm 1. A ruleset and sentence are accepted as input and generate questions as output. Within the ruleset, the similarity of each rule to the token list is calculated and finally, the winner is selected. Then the list of questions is generated using the winner rule applied to the token list. Out of the question lists, one best question is selected using bilingual evaluation understudy score (BLEU) automatic evaluation metrics.

Algorithm 1. Template-based question generation

```
Input: Ruleset, sentence
Output: Question
Begin
TokenList←Sentence
QuestionList=empty
Bestsimilarity=empty
Bestscore=empty
Foreach rule in ruleset
     begin
     sim=similarity (Rule, TokenList)
     if sim>Bestsimilarity {
     WinnerRule←Rule
     Bestsimilarity←sim}
     End
```

```
QuestionList=apply (WinnerRule, TokenList)
Foreach question in questionlist:
Begin
    Score=BLEU (sentence, Question)
    if score>Bestscore {
    Winner question←Question
    Bestscore← score}
End
Return WinnerQuestion
End
```

In our experiment, the rule set is constructed for the general domain by considering the most common English question patterns and the different structures of the sentences. Regarding the preprocessing phase the first step is tokenization. Tokenization is the mechanism by which a given expression is split into words or other significant elements called tokens. Another operations steps in the preprocessing phase are sentence segmentation, tokenization, part-of-speech (POS) tagging, and rule matching. Rule set construction and template matching is based on the POS tag feature vector of the tokens. Rules holds both sentence template and their question template. To apply on concrete sentence, the POS tag feature is determined for matching. Sentence template is given by list of POS tags with position index to differentiate the similar POS tags with in the sentence. e.g., [*NN1, VBZ1, VBN1, IN1, NN2*]. Question template is given by list of common question words and POS tags with position index to differentiate the similar POS tags with in the sentence e.g., [where, *NNS1, VBP1, VBN1, IN1, DT1, NN1*].

The next task is to evaluate each generated questions with the original sentences and return the best scorer question as a final result. Finally, the system generates a question with all possible constructed templates. Then the system automatically evaluates each generated question with the given sentence using the BLEU metric and takes the maximum score as the final output question. Example 1, let us assume the rule set contains the following three rules having different number of question templates.

Rule 1

{ST=[‘NNS1’,’VBP1’,’VBN1’,’IN1’,’DT1’,’NN1’]; QT= [’Where’+’VBP1’+’NNS1’+’ BN1’+’?’]}

Rule 2

{ST=[‘VBG1’,’NN1’,’VBZ1’,’NNS1’]; QT= [’Which’+’VBZ1’+’NNS1’+’?’]}

Rule 3

{ST=[‘NN1’,’VBZ1’,’VBN1’,’IN1’,’NN2’]; QT_1*= [’How’+NN1+’VBZ1’+’VBN1’+’?’];* QT_2*= [’NN1’+’VBZ1’+’VBN1’ +’IN1’+ ’what’+’?’];* QT_3*= [’Which’+ ’VBZ’ +’VBN’+’IN’+ ‘NN2’+’?’]}*

The input sentence is the following:

(*S=Limestone is formed by deposition)*;

In the first step we generate the token list and yielding the following list.

T= ‘NN1’,’VBZ1’,’VBN1’,’IN1’,’NN2’.

Based on the similarity calculation using edit distance, we have got the following similarity values for the rules:

$sim\ (T,ST_{r1}) =0.14,$
$sim\ (T,ST_{r2}) =0.08,$
$sim\ (T,ST_{r3}) =1,$

based on the best similarity score the winner is Rule 3. Using the substitutions, we have got the concrete variants for the question templates of the winner rule.

Q1=How limestone is formed? BLEU scores 0.55
Q2=Limestone is formed by what? BLEU scores 0.668
Q3=Which is formed by deposition? BLEU scores 0.56

Based on the BLEU score the winner question is “limestone is formed by what?”

3.2. MLP architecture

MLP is a supplement of feed-forward neural network and consists of three types of layers the input layer, output layer, and hidden layer [33]. The neural network architecture learns any function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Figure 2 ilustrated the training and prediction steps of MLP model. First, we read and pre-processing the dataset then train the model on the implemented model next train and make prediction on the testdata. We created datasets both manually and from the question-answer sentence composition (QASC) dataset to prepare the MLP training model. The QASC dataset [34] is a question-and-answer set that focuses on sentence composition. It includes a corpus of 17 million sentences and 9,980 multiple-choice questions regarding grade school science (8,134 train, 926 dev, 920 test).

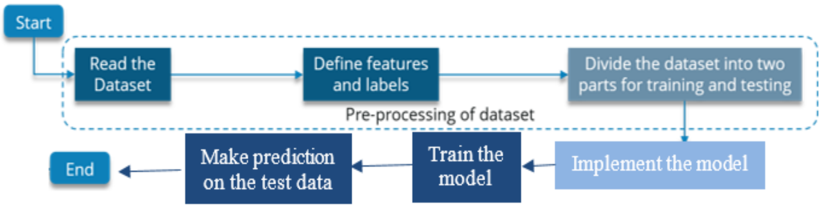

Start
Read the Dataset
Define features and labels
Divide the dataset into two parts for training and testing
Pre-processing of dataset
Implement the model
Train the model
Make prediction on the test data
End

Figure 2. Training and prediction steps of MLP model

We discovered several null values in the QASC dataset, as well as very long and useless sentences. Due to this, we try to preprocess and cleanup the dataset and we selected only the top 700 short and more meaningful sentence question pairs. In addition, we have constructed 300 sentence question pairs manually from general truth and common sentences, finally, we have to build 1000 sentence question pair datasets for our training. Table 1 shows sample sentence question pairs from our dataset.

Table 1. Sample sentence question pairs from our dataset

No	Sentence	Question
1	Ethiopia defeated Italy at the Battle of Adwa	Who won the battle of Adwa?
2	GERD is the largest dam in Africa	Which is the largest dam in Africa?
3	Beads of water can be formed by clouds	What type of water formation is formed by clouds?
4	Limestone is formed by deposition	What kind of rock is formed by deposition?
5	Ethiopia is a country comprised of 13 months	Which country has 13 months in a year?
6	Bacteria are found in soil	Where are bacteria found?
7	A fish can breathe in the water	Which can breathe in the water?

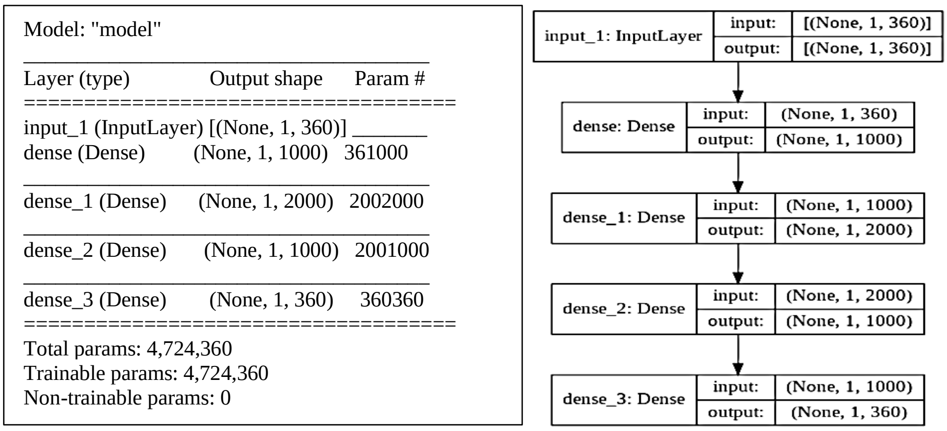
In our experiment, the first preprocessing step is converting the sentence into a sequence of phrases. Phrases are a combination of two or more words that can take the role of a noun, a verb, or a modifier in a sentence. In the English language, there are five phrase types i.e., noun phrase (NP), verb phrase (VB), adjective phrase (ADJP), adverb phrase (ADVP), and prepositional phrase (PP). We have used chunking to extract phrases from sentences. To construct the input matrix for the MLP model we build a vocabulary with a combination of English phrases and unique words that exist only in questions. Then we extend the vocabularity with the WH question words and the most frequeny unique words. In our test, we built up vocabulary containing 40 unique words. Sentence and question vector representation for MLP Training model illustrated in Figure 3.

$$\begin{bmatrix} sp1x0, sw1x1, sw1x2 \ \dots \ sp1x40 & qp1x0, qp1x1, qp1x2 \ \dots \ qp1x40 \\ sp2x0, sw2x1, sw2x2 \ \dots \ sp2x40 & qp2x0, qp2x1, qp2x2 \ \dots \ qp2x40 \\ sp3x0, sw3x1, sw3x2 \ \dots \ sp3x40 & qp3x0, qp3x1, qp3x2 \ \dots \ qp3x40 \\ \dots\dots & \dots\dots \\ sp10x0, sp10x1, sp10x2 \dots sp10x40 & qp10x0, qp10x1, qp10x2 \ \dots \ qp10x40 \end{bmatrix}$$

Figure 3. Sentence and question vector representation for MLP Training model

Then we convert the phrase tags of each sentence into vector form based on their vocabulary. Finally, the matrix of the training set is created using a one-hot encoding method. Based on our observation from our dataset the maximum length of phrases in a sentence or question is nine, which means the vector length of each sentence and question would be 40*9=360. To read all the vector form datasets, we have used a loop to combine in one array and form a none*1*360 matrix. Then the model starts to train the vector-matrix values of a single sentence train with every vector equivalent of the question phrase tag.

For our experiment, we defined the structure of a MLP network model. The model has 360 inputs, 3 hidden layers with 1000, 2000, and 1000 neurons, and an output layer with 360 output. Rectified linear activation functions are used in each hidden layer and a sigmoid activation function is used in the output layer. A plotted graph and input and output shapes of each layer illustrated in Figure 4. For regularization, we have tried dropout and there is no effect on the accuracy. A line plot of model classification accuracy for the training and validation datasets over training epochs is illustrated in Figure 5. The line plot demonstrates that the model learns the problem quickly, achieving an accuracy of about 80% in roughly 25 epochs rather than the 100 epochs. The line plot also illustrates that throughout training, train and test performance are comparable.



```
Model: "model"
_________________________________________________
Layer (type)            Output shape      Param #
===================================
input_1 (InputLayer) [(None, 1, 360)] _______
dense (Dense)        (None, 1, 1000)   361000
_________________________________________________
dense_1 (Dense)      (None, 1, 2000)  2002000
_________________________________________________
dense_2 (Dense)      (None, 1, 1000)  2001000
_________________________________________________
dense_3 (Dense)      (None, 1, 360)    360360
===================================
Total params: 4,724,360
Trainable params: 4,724,360
Non-trainable params: 0
```

Figure 4. A model and plotted a graph of our MLP neural network

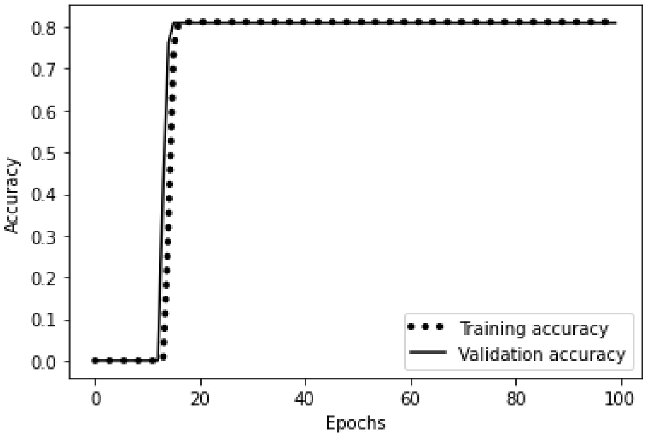


Figure 5. Accuracy on train and validation datasets

4. RESEARCH METHOD

We developed our proposed system using google colaboratory, or "colab" for short, which allows us to write and execute python in our browser with no additional configuration. Colab notebooks run code on google's cloud servers, allowing you to take advantage of Google hardware, such as GPUs and TPUs, regardless of your machine's capabilities. The researchers divided the dataset into a training set (90 percent) and a testing set (10 percent) using random sampling techniques. After we have implemented our proposed system, we need to measure and compare their efficiency. According to our observation, most scholars do not come into consciences of which methodologies use for the evaluation techniques of question generation. It is hard to quantify the generated questions "good" because good questions tend to be significant, syntactically correct, semantically sound, and natural. As a result, recent QG research tends to utilize human evaluation. However, human evaluation can be labor-intensive, time-consuming, inconsistent, and hard to reproduce. Due to these, researchers [35], [36] still use automatic evaluation metrics even though studies have shown that automatic evaluation metrics do not correlate well with fluency and coherence.

In our evaluation methodology, we used human raters to blindly comparing automatically-generated questions with human-generated (golden questions) rating (1-5) marks for all testing questions and BLEU and ROUGE automatic evaluation metrics. The BLEU for short, is a metric for evaluating a generated sentence to a reference sentence. BLEU was originally created to measure the quality of machine translation with respect to human translation [37]. It computes an N-gram precision difference between the two sequences, as well as a penalty for machine sequences being shorter than human sequences. A perfect match receives a 1.0 score, whereas a perfect mismatch receives a 0.0 value. The most you can do is get a 0.6 or 0.7 on the scale. This score was created primarily to assess the accuracy of automatic machine translation systems' predictions [35]. On a set of references, BLEU calculates the average n-gram precision. A BLEU-n score is a BLEU score that has been calculated using up to n-grams.

The other metric employed is recall-oriented understudy for gisting evaluation (ROUGE), a set of evaluation metrics proposed in the context of automatic summarization [36]. ROUGE is a collection of metrics rather than a single metric. ROUGE-N is the one that is most likely to be used. The N in ROUGE-N stands for the n-gram that we're employing. We would measure the match rate of unigrams in ROUGE-1, and bigrams between our model output and reference in ROUGE-2. We'll calculate the ROUGE recall, precision, or F1 score once we've determined which N to utilize [38]. The last metric is ROGUE-L [37], which is based on the length of the longest common subsequence (LCS) between our model output and the reference sequence. It calculates the final as the F-measure of these values above, using the fractional length of LCS over sequence length as precision/recall for one and vice versa for the other. To implement these metrics we have used the Python rouge library. To further inspect the capabilities of our proposed QG models, we also perform human evaluation on our template-based and MLP based question generation models.

5. RESULTS AND ANALYSIS

For human evaluation, we have prepared 100 sentences randomly from different sources and we asked three annotators to score them on the scale of [1-5] independently, with the following three metrics: Fluency: whether a question is grammatical and fluent. Relevancy: whether the question is semantic relevant to the passage. Answerability: whether the question can be answered by the right answer [39].

Table 2 shows the sample sentences and questions generated by human, template-based system and MLP-based system. The questions generated using the proposed system are evaluated using both automatic metrics and human evaluators regarding the gold questions. The survey was executed on google forms and evaluated by 10 fluent English speakers. Before starting the survey, the evaluators were informed about the purpose of the study and the questionnaire.

Table 2. Sample test sentence question pair with generated questions

No	Sentence	Human Generated Question	Template Based Generated Question	MLP based Generated Question
1	Ethiopia defeated Italy at the Battle of Adwa	Who won the battle of Adwa?	Who defeated Italy at battle Adwa?	What Ethiopia defeated the battle?
2	Hearing is the fastest human sense	Which is the fastest human sense?	What does Hearing do fastest?	What is the fastest human sense?
3	The fastest bird is the Peregrine falcon.	What is the fastest bird?	What does the bird is the?	What is the peregrine falcon?
4	Dragonflies are one of the fastest insects	What are the fastest insects?	Who are the fastest of the insects?	What are the fastest insects?

However, human evaluators rate all questions generated by humans and the proposed system as shown in Table 3. The BLEU and ROUGE automatic metrics evaluation result is displayed in Table 4. A BLEU implementor's is to compare the candidate's n-grams to the reference's n-grams and count the number of matches. Position has no impact on these matches. The higher the number of matches, the better the candidate [35].

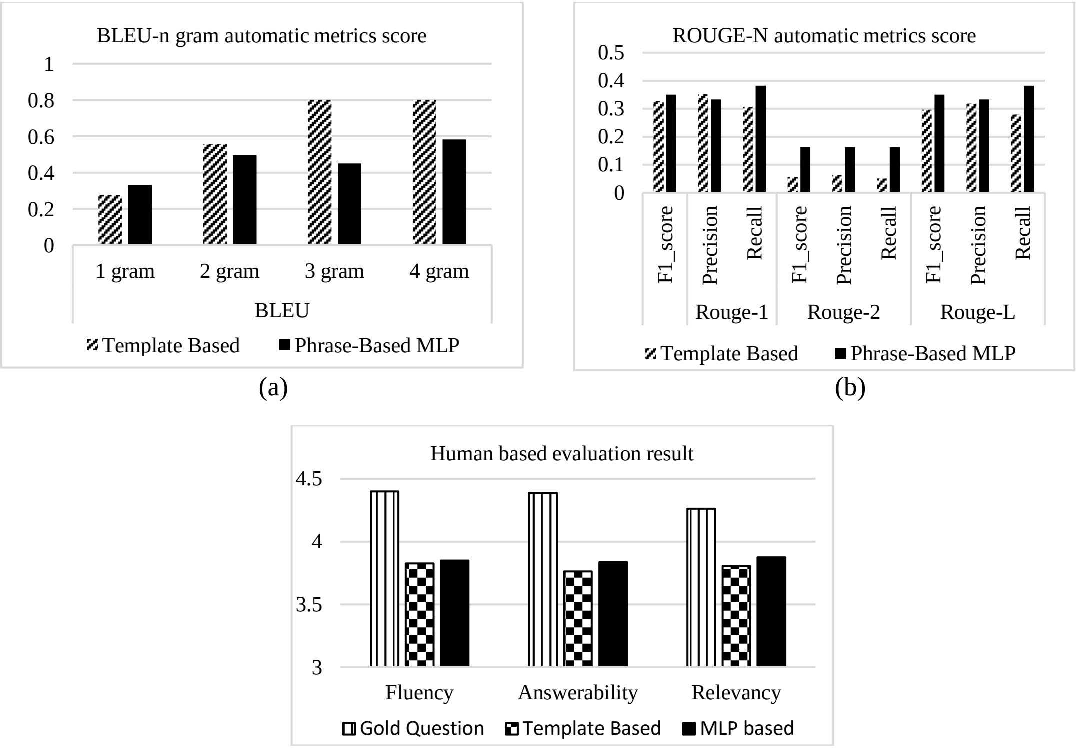
Table 3. Human-based evaluation result

	Fluency	Answerability	Relevancy
Gold question	4.4	4.387	4.262
Template based	3.825	3.762	3.805
MLP based	3.85	3.837	3.875

Table 4. Automatic metrics evaluation result

		Template based AQG model	Phrase-based MLP AQG model
BLEU	1 gram	0.2778	0.331
	2 gram	0.556	0.496
	3 gram	0.8	0.451
	4 gram	0.8	0.582
Rouge-1	F1_score	0.327160714	0.350872356
	Precision	0.351731602	0.333549784
	Recall	0.307359307	0.381818182
Rouge-2	F1_score	0.056565655	0.163636362
	Precision	0.063636364	0.163636364
	Recall	0.051515152	0.163636364
Rouge-L	F1_score	0.296645774	0.350872356
	Precision	0.318398268	0.333549784
	Recall	0.279220779	0.381818182

The summary of test results has presented in Figure 6 with a better visualization. As Figure 6 shows, questions generated by the template-based system have got the highest score in BLEU 3 and 4-gram quality measures, but the weaker results for the rest metrics. Figure 6 shows that in all ROUGE-N automatic metrics phrase-based generated questions score better results than template-based generated. We have evolved human evaluators who consider the Fluency, Answerability, and relevancy of each sentence with regard to the questions. In this evaluation, all generated question types i.e., gold question, template-based, MLP based need to be evaluated. The evaluation result presented in Table 4 shows that MLP based approach has definitely better results than the template-based. As shown in Figure 6 gold questions have score best result in human evaluation.



(a)

(b)

(c)

Figure 6. Comparison of template-based and phrase-based QG using (a) BLEU-n gram, (b) ROUGE-N, and (c) automatic evaluation

According to the overall evaluation result, MLP based approach definitely got a better score. In addition, we observed that humans can understand and answer the automatically generated questions. From this, we notice that our proposed MLP based system is more encouraging than a template-based approach.

6. CONCLUSION

AQG is a key module in Intelligent Tutoring Systems. The paper presents an efficiency comparison of two generation methods on a closed domain. The first method is the template-based approach and the second is the neural network-based method. The quality of the generated questions was evaluated by both automatic metric scores (BLEU and ROUGE) and by human experts. The performed test experiences show that both methods can provide a good, 80%-88% accuracy in the test generation compared to human generated questions. Based on the test results, the neural network-based method using NLP dominates the classic template-based approach not only in the open world domain but also in the closed word domain.

ACKNOWLEDGEMENTS

The research reported here was carried out as part of the EFOP-3.6.1-16-2016-00011 “younger and renewing university–innovative knowledge city–institutional development of the University of Miskolc aiming at intelligent specialization” project implemented in the framework of the Széchenyi 2020 program. The realization of this project is supported by the European union, co-financed by the European social fund.

REFERENCES

[1] J. Carbonell, “AI in CAI: an artificial-intelligence approach to computer-assisted instruction,” *IEEE Transactions on Man Machine Systems*, vol. 11, no. 4, pp. 190–202, Dec. 1970, doi: 10.1109/TMMS.1970.299942.

[2] A. H. Muhammad and D. Ariatmanto, “Understanding the role of individual learner in adaptive and personalized e-learning system,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 6, pp. 3313–3324, Dec. 2021, doi: 10.11591/eei.v10i6.3192.

[3] J. Cao, T. Yang, I. K.-W. Lai, and J. Wu, “Student acceptance of intelligent tutoring systems during COVID-19: The effect of political influence,” *The International Journal of Electrical Engineering & Education*, p. 002072092110032, Mar. 2021, doi: 10.1177/00207209211003270.

[4] Y. Long and V. Aleven, “Skill diaries: improve student learning in an intelligent tutoring system with periodic self-assessment,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7926 LNAI, 2013, pp. 249–258.

[5] K. E. Ehimwenma and S. Krishnamoorthy, “Design and analysis of a multi-agent e-learning system using prometheus design tool,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, p. 9, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp9-23.

[6] K. E. Boyer and P. eds Piwek, *QG2010 : The Third Workshop on Question Generation*. 2010.

[7] S. Soni, P. Kumar, and A. Saha, “Automatic question generation: a systematic review,” *SSRN Electronic Journal*, 2019, doi: 10.2139/ssrn.3403926.

[8] L. K. Walelign Tewabe Sewunetie, Ghanim Hussein Ali Ahmed, “The development and analysis of extended architecture model for intelligent tutoring systems,” *Gradus*, vol. 6, no. 4, pp. 128–138, 2019.

[9] F. S. Cohen, “What is a question?,” *Monist*, vol. 39, no. 3, pp. 350–364, 1929, doi: 10.5840/monist192939314.

[10] A. Echihabi and D. Marcu, “A noisy-channel approach to question answering,” in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL ’03*, 2003, vol. 1, pp. 16–23, doi: 10.3115/1075096.1075099.

[11] V. Rus and J. Lester, *The 2nd workshop on question generation*. 2009.

[12] N. Karamanis, L. A. Ha, and R. Mitkov, “Generating multiple-choice test items from medical text,” in *Proceedings of the Fourth International Natural Language Generation Conference on - INLG ’06*, 2006, p. 111, doi: 10.3115/1706269.1706291.

[13] J. Mostow and W. Chen, “Generating instruction automatically for the reading strategy of self-questioning,” in *Frontiers in Artificial Intelligence and Applications*, 2009, vol. 200, no. 1, pp. 465–472, doi: 10.3233/978-1-60750-028-5-465.

[14] T. Liu, B. Wei, B. Chang, and Z. Sui, “Large-scale simple question generation by template-based seq2seq learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10619 LNAI, 2018, pp. 75–87.

[15] H.-T. Zheng, J. Han, J. Chen, and A. Sangaiah, “A novel framework for automatic Chinese question generation based on multi-feature neural network model,” *Computer Science and Information Systems*, vol. 15, no. 3, pp. 487–499, 2018, doi: 10.2298/CSIS171121018Z.

[16] O. Keklik, T. Tugular, and S. Tekir, “Rule-based automatic question generation using semantic role labeling,” *IEICE Transactions on Information and Systems*, vol. E102.D, no. 7, pp. 1362–1373, Jul. 2019, doi: 10.1587/transinf.2018EDP7199.

[17] M. H. Fattoh, Ibrahim E and Aboutabl, Amal E and Haggag, “Semantic question generation using artificial immunity,” *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 1, pp. 1–8, 2015.

[18] M. Agarwal, R. Shah, and P. Mannem, “Automatic question generation using discourse cues,” in *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, 2011, pp. 1–9.

[19] G. D. Anderson, “Auxiliary verb constructions,” in *OUP Oxford*, 2006.

[20] C. Jouault and K. Seta, “Building a semantic open learning space with adaptive question generation support,” in *Proceedings of the 21st International Conference on Computers in Education, ICCE 2013*, 2013, pp. 41–50.

[21] M. Heilman, “Automatic factual question generation from text,” Doctoral dissertation, School of Computer Science, Carnegie Mellon University, US, 2011.

[22] Y. Chali and S. A. Hasan, “Towards topic-to-question generation,” *Computational Linguistics*, vol. 41, no. 1, pp. 1–20, Mar. 2015, doi: 10.1162/COLI_a_00206.

[23] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-To-end dialogue systems using generative hierarchical neural network models," *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, vol. 30, no. 1, pp. 3776–3783, Mar. 2016, doi: 10.1609/aaai.v30i1.9883.

[24] Q. Zhou *et al.*, "National CCF conference on natural language processing and chinese computing," in *arXiv preprint arXiv:1704.01792*, 2017, pp. 662--671, doi: National CCF Conference on Natural Language Processing and Chinese Computing.

[25] B. Das, M. Majumder, S. Phadikar, and A. A. Sekh, "Automatic question generation and answer assessment: a survey," *Research and Practice in Technology Enhanced Learning*, vol. 16, no. 1, p. 5, Dec. 2021, doi: 10.1186/s41039-021-00151-1.

[26] M. Blšták and V. Rozinajová, "Automatic question generation based on sentence structure analysis using machine learning approach," *Natural Language Engineering*, vol. 28, no. 4, pp. 487–517, Jul. 2022, doi: 10.1017/S1351324921000139.

[27] M. Sarrouti, A. Ben Abacha, and D. Demner-Fushman, "Goal-driven visual question generation from radiology images," *Information*, vol. 12, no. 8, p. 334, Aug. 2021, doi: 10.3390/info12080334.

[28] N. Duan, D. Tang, P. Chen, and M. Zhou, "Question generation for question answering," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 866–874, doi: 10.18653/v1/D17-1090.

[29] B. Liu, "Neural question generation based on Seq2Seq," in *Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence*, Apr. 2020, pp. 119–123, doi: 10.1145/3395260.3395275.

[30] C.-Y. Chen, H.-C. Liou, and J. S. Chang, "FAST," in *Proceedings of the COLING/ACL on Interactive presentation sessions -*, 2006, pp. 1–4, doi: 10.3115/1225403.1225404.

[31] A. Hoshino and H. Nakagawa, "Assisting cloze test making with a web application," in *Proceedings of Society for Information Technology & Teacher Education International Conference 2007*, 2007, pp. 2807–2814.

[32] E. Forsbom, "Training a super model look-alike: featuring edit distance, N-gram occurrence, and one reference translation," in *In Proceedings of the Workshop on Machine Translation Evaluation. Towards Systemizing MT Evaluation*, 2003, pp. 29–36.

[33] M. Perceptron and S. Abirami, *The digital twin paradigm for smarter systems and environments: the industry use cases 3.1 multi layer perceptron using goals in model-based reasoning*. Elsevier, 2020.

[34] T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal, "QASC: A dataset for question answering via sentence composition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8082–8090, Apr. 2020, doi: 10.1609/aaai.v34i05.6319.

[35] B. Babych, "Automated MT evaluation metrics and their limitations," *Tradumàtica: tecnologies de la traducció*, no. 12, p. 464, Dec. 2014, doi: 10.5565/rev/tradumatica.70.

[36] P. Nema and M. M. Khapra, "Towards a better metric for evaluating question generation systems," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018, pp. 3950–3959, doi: 10.18653/v1/d18-1429.

[37] J. Singh and Y. Sharma, "Encoder-decoder architectures for generating questions," *Procedia Computer Science*, vol. 132, pp. 1041–1048, 2018, doi: 10.1016/j.procs.2018.05.019.

[38] J. Briggs, "The ultimate performance metric in NLP." towardsdatascience.com. 2021. https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460. l (accessed Jul. 12, 2022).

[39] X. Jia, W. Zhou, X. Sun, and Y. Wu, "EQG-RACE: examination-type question generation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, pp. 13143–13151, May 2021, doi: 10.1609/aaai.v35i14.17553.

BIOGRAPHIES OF AUTHORS

Walelign Tewabe Sewunetie received the B.Sc. degree in computer science from the Mekelle University, Ethiopia, and the M.Sc. degree in computer science from the Arba Minch University, Ethiopia. He was the Deputy registrar director of Debre Markos Institute of Technology, Debre Markos University, Ethiopia. He was also acting as the Managing Director of Debre Markos Institute of Technology at Debre Markos University, Ethiopia from 2018 to August 2019. He is currently a Ph.D. candidate at the University of Miskolc, Hungary. His research interests include soft computing, machine learning, NLP, and intelligent tutoring systems. He can be contacted at email: walelig ntewabe@gmail.com.

László Kovács is a full professor at the University of Miskolc, Hungary, Institute of Information Sciences. He is the Department head of Software Engineering. His main research area includes database and knowledge base modeling, concept lattice structures, discrete optimizations, and machine learning algorithms in NLP. He has authored or coauthored more than 194 publications: 109 conference proceedings and 85 journals. He has supervised more than 7 Ph.D. students. He can be contacted at email: kovacs@iit.uni-miskolc.hu.