DOI: 10.24996/ijs.2025.66.12.36

ISSN: 0067-2904

Arabic Question Generation Based on the Arabic Language's Structural Characteristics

Husam N. Fawzi[1]*, Suhad A. Yousif[1], Layth Shakir Mahmood[2]
[1]Computer Science Department, College of Science, Al-Nahrain University, Baghdad, Iraq
[2]Department of History, College of Arts, University of Baghdad, Baghdad, Iraq

Received: 24/6/2024 Accepted: 17/11/2024 Published: 30/12/2025

Abstract

Automatic question generation is a popular field for many researchers because it is vital in reducing the time and effort of educators in many fields. Still, the main focus is on the English language, and the tools for question generation in Arabic still need to be developed. This paper seeks to overcome this limitation by proposing a method to automate or semi-automate the task of question generation for the Arabic language focused on history topics, using natural language processing's rule-based approach and employing the Arabic language's structural characteristics. Our method successfully generated fill-in-the-blanks MCQ with distractors, and True/False questions with the correct answer, offering the capability of automatic grading without any reliance on external sources or pre-trained models. The proposed method was tested on an Arabic article that contained historical information about the Sumerian civilization and used to generate 100 fill-in-the-blanks MCQ and 100 True/False questions. A manual evaluation by history experts was performed on the generated questions. The generated MCQ achieved (82%) of acceptable questions, and the True/False questions achieved (70%). The findings in this research provided a promising approach toward automating the question generation for the Arabic language while minimizing the required computational resources.

Keywords: Artificial Intelligence, Natural Language Processing, NLP Rule-Based Systems, Linguistic Rules in Arabic, Automatic Question Generation.

توليد الأسئلة باللغة العربية بناءً على الخصائص الهيكلية للغة العربية

حسام نمير عبد السلام فوزي[1]*, سهاد عبد الرحمن يوسف[1], ليث شاكر محمود[2]
[1]علوم الحاسوب, كلية العلوم, جامعة النهرين, بغداد, العراق
[2]قسم التاريخ, كلية الآداب, جامعة بغداد, بغداد, العراق

الخلاصة

ان أتمتة توليد الأسئلة هو مجال شائع بين العديد من الباحثين الأكاديميين لأنه مهم في تقليل الوقت والجهد المطلوبين من المعلمين في العديد من المجالات. وان التركيز الأساسي في هذا المجال هو للغة الإنكليزية، مع بقاء الأدوات المستخدمة في توليد الأسئلة للغة العربية غير مطورة. في هذا البحث نعالج هذه المحددات، باقتراح طريقة لأتمتة أو شبه أتمتة توليد الأسئلة للغة العربية، للنصوص المختصة بمادة التاريخ،

*Email: husam.namir@nahrainuniv.edu.iq

باستخدام نهج قائم على القواعد في معالجة اللغة الطبيعية واستغلال الخصائص الهيكلية للغة العربية. نجحت طريقتنا في توليد أسئلة ملء الفراغات بنمط اختيارات متعددة وأسئلة صح/خطأ مع الإجابة الصحيحة، مما يوفر إمكانية التصحيح الآلي، دون الاعتماد على مصادر خارجية أو نماذج مدربة مسبقًا. تم اختبار الطريقة المقترحة على مقال عربي يحتوي على معلومات تاريخية حول الحضارة السومرية، وتم استخدام الطريقة لتوليد 100 سؤال ملء الفراغات بنمط اختيارات متعددة و100 سؤال صح/خطأ. تم إجراء تقييم يدوي من قبل اخصائيين في التاريخ على الأسئلة المولدة والاختيارات المتعددة بشكل فردي، حققت أسئلة الاختيارات المتعددة نسبة قبول بلغت (82%)، في حين حققت أسئلة الصح/خطأ نسبة (70%). توفر النتائج في هذا البحث نهجًا واعدًا نحو أتمتة توليد الأسئلة للغة العربية مع تقليل الموارد الحاسوبية المطلوبة.

1. Introduction

How do you evaluate your level of understanding of a lesson? The simple answer is to take a test, and the grade will give you the answer. For that reason, a lot of effort has been put into developing tests and choosing the best type of questions that will help produce a fair evaluation of the level of understanding, as explained in the studies of [1]-[3].

Automatic question generation (AQG) is a branch of natural language processing (NLP) that focuses on automating question generation by developing algorithms and systems capable of conducting such tasks without any human intervention.

The task of (AQG) is an interesting point for many researchers due to the significant role that it plays in both educational and self-assessment domains, as demonstrated in the studies of [4]-[6], that answering questions from a given course is the ideal way of evaluating the learning process. AQG reduces the time and effort that an educator might spend on manually creating questions from a textbook or a lecture and can also offer the possibility of automatic grading.

The main problem that we address in this paper is that more research materials related to AQG for the Arabic language need to be made. However, many new modern Artificial intelligence (AI) language models, such as (ChatGPT) or pre-trained models [7, 8], can successfully solve this task. Still, those artificial intelligence models need to be more transparent about their work as they use proprietary algorithms that are not available to the researcher's community or the public. Also, they have been trained on massive amounts of data, requiring huge computational resources for them to work.

In this paper, we introduce a method to AQG for the Arabic language focused on history topics, using (NLP) 's rule-based approach and employing the Arabic language's structural characteristics to generate fill-in-the-blanks MCQ and True/False questions.

Our method can work on any given topic related to historical information without prior knowledge or external sources but relies on the input text to generate the questions. It also automates the grading as it generates the correct answer for each generated question. Furthermore, our method can be counted as a contribution towards the green initiative by reducing our carbon footprint as it requires fewer computational resources and will consume less energy during the work compared to the other methods that use machine learning models or similar approaches.

The remainder of this paper is organized in the following order: Section 2 summarizes the related work that has been done for AQG. In section 3, we describe our methodology. Section

4 discusses how we evaluated our method. In section 5, we present the results of our evaluation. Finally, section 6 concludes our paper and mentions some future scopes that can enhance our work.

2. Related Work

A lot of effort has been put into the field of AQG, especially for the English language [9]-[11], which included many methods for question generation using NLP techniques such as syntax-based, semantic-based, and template-based; however, very few researchers have tackled this field for the Arabic language.

In their research, Riken Shah [12] generated MCQ questions from a given text, specifically for physics topics. The system uses NLP to generate the questions with the correct answers and distractors. The author uses Wikipedia articles as the knowledge source to generate the distractors, using the Inverse Document Frequency (IDF) measure to rank the keywords and the Context-Based Similarity approach to generate the distractors. Still, this approach depends on pre-existing knowledge from Wikipedia articles.

In the study of [13], combined NLP syntax-based and semantic-based methods were proposed to develop an AQG system. Still, the proposed system only generated WH-question (who, what, when, where, why, which, whom), and it did not generate other forms of questions such as MCQ or True/False questions.

According to research conducted by [14], multiple views of text from different parsers were used to form a tree structure, which enhances the question generation process. The author uses NLP pattern matching, in which the text is searched for pattern form that can be used as a question, in contrast to the study of [15] that infuses natural language understanding (NLU) into the process of natural language generation (NLG); by analyzing the semantic content of each cluster in each sentence then using a question template to understand what the sentence is trying to communicate to improve the quality of the generated question.

An AQG system proposed by [16] generates descriptive and factoid questions using relative adverbs and pronouns from English sentences. The system is syntax-based and uses information from the input sentences obtained from a dependency parser.

A prototype AQG system was developed by [17]. It uses classification and clustering techniques for NLP. The methodology involves collecting words that are clustered from the input text and employing neural network classification to select the most suitable distractors for the question that is being generated.

In the study of [18], an ontology-based approach was used for AQG in a Virtual chemical laboratory. This approach helped generate distractors and refine the generated question, but it requires pre-existing knowledge.

[19] presented a method for AQG to use in the Thai language. The author used a syntax-based method to generate the questions and employed Google Translate API to generate the distractors from the WordNet library. This method relies on Google's translation capabilities and might generate incompatible distractors. First, the answer is translated from Thai to English, all the related distractors are found in WordNet, and then finally, those distractors are translated back to Thai.

The paper [20] is one of the few that addresses the issue of no AQG for the Arabic natural language process (ANLP) community [21,22]. The author used Templates based on the REGEX languages. The main drawbacks are that this approach only generates WH-questions and does not generate MCQ and True/False questions. Also, it needs to generate the correct answer, so there are no automatic grading capabilities in this method.

In this paper, we propose a method that addresses the main limitation of AQG for the Arabic language. Our method generates MCQ with the distractors and True/False questions with the correct answer, allowing those questions to be automatically graded and eliminating the need for manual assessment of the answers.

3. Methodology

In this section, we will provide a detailed explanation of the proposed method, including procedural steps for each part.

The proposed method, illustrated in Figure 1, takes Raw text as input and then applies the data preprocessing algorithm. After that, it will generate MCQ or True/False questions (based on our decision) by applying the respective algorithms, and finally, the output will either be the generated MCQ or True/False questions.

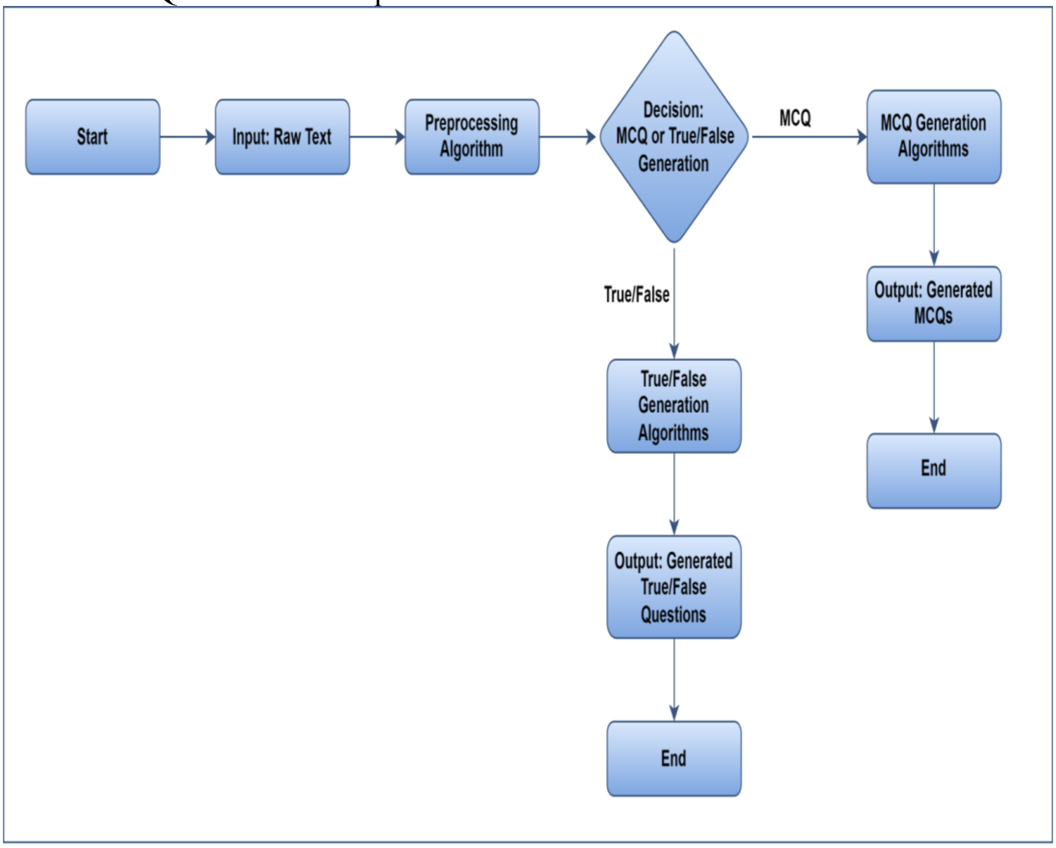

Figure 1: Methodology for Generating MCQs and True/False Questions from Raw Text

The remainder of this section will include the detailed procedural steps for each algorithm mentioned above.

3.1 Data Preprocessing:

The first step is data preprocessing, in which we take raw text as the input and produce the lists of identified proper nouns and numbers in the given text as well as the Tokenized sentences as the output; the procedural steps are further explained in algorithm 1.

Algorithm 1: Data Preprocessing.	
Input:	InputText (raw text).
Output:	SentenceList (Processed Sentences), NounList, and NumberList.
Step1:	Initialize empty lists NounList and SentenceList.
Step2:	Initialize empty list NumberList with two fields (Number, NumberType).
Step3:	Tokenize InputText into sentences using regular expression (RE) to detect sentence boundaries (period, question mark, exclamation mark, new line). Store the sentences in SentencesList.
Step4:	For each sentence in SentencesList, do the following: a. If the sentence length is less than a certain threshold of words (we chose ten words as our threshold), remove the sentence from SentencesList, as they donate headings or titles within the original text rather than useful information. b. Remove any numbers (digits or written form) found at the beginning of the sentence, as they denote sentence order rather than useful information.
Step5:	Utilize Stanza Library for Named Entity Recognition (NER) on InputText to identify proper nouns. Store the identified proper nouns in NounList.
Step6:	Utilize Stanza Library for Part-of-Speech (POS) tagging on InputText to identify numbers. Record whether the number is in digit or written form and store it in NumberList.
Step7:	Shuffle the sentences in SentencesList to ensure randomness for the generated questions.

We decided to use Named Entity Recognition (NER) instead of Part-of-Speech (POS) tagging to identify proper nouns in Arabic texts. This decision was based on our testing, which showed that POS tagging returned both proper and common nouns, making it difficult to distinguish between them. On the other hand, NER consistently identified proper nouns in Arabic texts. This is important for generating coherent Arabic questions with relevant distractors. However, we found that POS tagging was useful for identifying Arabic numbers, whether in digit or written format.

3.2 Fill-in-the-blanks MCQ Generation:

We mainly have two rules for generating MCQs: proper noun and number replacement. For each rule, we select the noun or the number as the keyword and choose the distractors from the previously extracted nouns or numbers during the preprocessing step.

3.2.1 Proper Noun Replacement for Fill-in-the-Blanks MCQ:

This rule scans each sentence for proper nouns, then selects the found noun as the keyword, as further explained in algorithm 2.

Algorithm 2: Proper Noun Replacement for Fill-in-the-Blanks MCQ.	
Input: SentenceList (list of sentences), NounList (list of proper nouns). **Output:** GeneratedQuestionsList (list of generated questions and four choices with the correct one).	
Step1:	Calculate the total number of nouns in NounList. If the total is below a certain threshold, exit the algorithm, as there are not enough distractors within the input text.
Step2:	For each sentence in SentenceList, do: a. Loop through NounList. b. If a proper noun from NounList is found in the sentence, replace the first found noun with a blank (_____). c. Add the modified sentence to the generated questions list. d. Set the replaced noun as the correct choice in the correct choice field. e. Select three other distractors from NounList and add them to the multiple-choice field of the generated question in the generated questions list.

3.2.2 Number Replacement for Fill-in-the-Blanks MCQ:

This rule scans each sentence for numbers that are either written or digits, then selects the found number as the keyword, as further explained in algorithm 3.

Algorithm 3: Number Replacement for Fill-in-the-Blanks MCQ.	
Input: SentenceList (list of sentences), NumberList (list of numbers) **Output:** GeneratedQuestionsList (list of generated questions and four choices with the correct one).	
Step1:	For each sentence in SentenceList, do: a. Loop through NumberList. b. If a number from NumberList is found in the sentence, replace the first found number with a blank (_____). c. Add the modified sentence to the generated questions list. d. Set the replaced number as the correct choice in the correct choice field. e. Determine the type of the replaced number (written or digit). f. If the replaced number is written, select three other numbers from NumberList with the same type as distractors. g. If the replaced number is a digit or there are not enough written numbers in NumberList, generate three random numbers (digits) as distractors. h. Add the replaced number (correct choice) and the selected distractors to the multiple-choice field associated with the generated question to the generated questions list.

3.3 True/False Questions Generation:

To control the ratio of the true and false in the generated questions, we are introducing a stochastic element for each True/False question generation algorithm; by generating a random Boolean variable from predefined weights, we can ensure that not all the generated questions are exclusively true or false. This probabilistic approach allows us to maintain a balanced ratio of true and false questions.

3.3.1 Proper Noun Replacement for True/False Questions:

This rule scans each sentence for proper nouns and replaces it with another from a different sentence to generate a false question, as further explained in algorithm 4.

Algorithm 4: Proper Noun Replacement for True/False Questions.	
Input: SentenceList (list of sentences), NounList (list of proper nouns). **Output:** GeneratedQuestionsList (list of generated true/false questions with replaced proper nouns and correct choice).	
Step1:	Calculate the total number of nouns in NounList.
Step2:	If the total number of nouns is above a certain threshold, continue to the next step; otherwise, exit the algorithm.
Step3:	For each sentence in SentenceList, do: a. Introduce a stochastic element by generating a random Boolean value based on predefined weights to determine whether the generated question should be true or false. b. If the generated Boolean value is false, add the original sentence to the generated questions list, set the correct choice as true, and go to the next sentence. c. Otherwise, continue to the next steps. d. Loop through NounList. e. If a proper noun from NounList is found in the sentence, replace the first found noun with another proper noun from NounList. f. Add the modified sentence to the generated questions list. g. Set the correct choice for the question as false since the proper noun was replaced.
Step4:	Append the phrase "(صحيح ام خطأ؟)" (meaning "true or false?") to each generated question in the generated questions list.

3.3.2 Number Replacement for True/False Questions:

This rule scans each sentence for numbers and replaces it with another one that is either randomly generated or selected from a different sentence to generate a false question, as further explained in algorithm 5.

Algorithm 5: Number Replacement for True/False Questions.	
Input: SentenceList (list of sentences), NumberList (list of numbers). **Output:** GeneratedQuestionsList (list of generated true/false questions with replaced numbers and correct choice).	
Step1:	For each sentence in SentenceList, do: a. Introduce a stochastic element by generating a random Boolean value based on predefined weights to determine whether the generated question should be true or false. b. If the generated Boolean value is false, add the original sentence to the generated questions list, set the correct choice as true, and go to the next sentence. c. Otherwise, continue to the next steps. d. Loop through NumberList. e. If a number from NumberList is found in the sentence, proceed to the next steps; otherwise, continue to the next sentence. f. Determine the type of number found (written or digit). g. If the found number is written and there are other written numbers in NumberList, select another written number from NumberList and replace the found number with it. h. If the found number is a digit or there are no other written numbers in NumberList, randomly generate a new number (digits) and replace the found number with it. i. Add the modified sentence to the generated questions list. j. Set the correct choice for the question as false since the number was replaced.
Step2:	Append the phrase "(صحيح ام خطأ؟)" (meaning "true or false?") to each generated question in the generated questions list.

3.3.3 Negation Particles Removal for True/False Questions:

This rule scans each sentence for negation particles and removes it to generate a false question, as further explained in algorithm 6.

Algorithm 6: Negation Particles Removal for True/False Questions.	
Input: SentenceList (list of sentences). **Output:** GeneratedQuestionsList (list of generated true/false questions with removed negation particles and correct choice).	
Step1:	For each sentence in SentenceList, do: a. Introduce a stochastic element by generating a random Boolean value based on predefined weights to determine whether the generated questions should be true or false. b. If the generated Boolean value is false, add the original sentence to the generated questions list and set the correct choice as true. c. If the generated Boolean value is true, proceed with the next steps. d. Identify if the sentence contains an Arabic negation particle (لا, لم, ما, ليس, ماالي, لن, لاشك, وليس) using word boundary RE. e. If a negation particle is found, remove it from the sentence. f. Add the modified sentence to the generated questions list. g. Set the correct choice for the question as false since the negation particle was removed.
Step2:	Append the phrase "(صحيح ام خطا؟)" (meaning "true or false?") to each generated question in the generated questions list.

3.3.4 Adjective Antonym Replacement for True/False Questions:

This rule scans each sentence for adjectives and replaces them with their antonyms to generate a false question, as further explained in algorithm 7.

Algorithm 7: Adjective Antonym Replacement for True/False Questions.	
Input: SentenceList (list of sentences). **Output:** GeneratedQuestionsList (list of generated true/false questions with replaced adjectives and correct choice).	
Step1:	For each sentence in SentenceList, do: a. Introduce a stochastic element by generating a random Boolean value based on predefined weights to determine whether the generated questions should be true or false. b. If the generated Boolean value is false, add the original sentence to the generated questions list and set the correct choice as true. c. If the generated Boolean value is true, proceed with the next steps. d. Utilize the Stanza library POS tagging to identify adjectives in the sentence. e. If an adjective is found, replace it with its antonym from the Arabic-LT library. f. Otherwise, continue to the next sentence. g. Add the modified sentence to the generated questions list. h. Set the correct choice for the question as false since the adjective was replaced by its antonym.
Step2:	Append the phrase "(صحيح ام خطا؟)" (meaning "true or false?") to each generated question in the generated questions list.

3.3.5 No Modification for True/False Questions:

Suppose no modifications are made to the sentences when the stochastic element returns false. In that case, the above algorithms will add the original sentences to the (GeneratedQuestionsList) and append the phrase "(صحيح ام خطا؟)" to each added sentence. Also, the correct choice will be true since no modifications are implemented in the sentences.

One last thing to mention about our proposed method is that for the proper noun and number replacement rules in both the MCQ and True/False questions, the choice to loop through the noun list and number list to identify if the sentence contains a proper noun or a number instead of utilizing NER and POS tagging again was to reduce the execution time as NER and POS tagging requires a substantial amount of computing resources, by doing so, we have significantly decreased the execution time for our method.

4. Evaluation

We used an Arabic article titled "Sumer Civilization," which can be found on the website (ibelieveinsci.com). The article provided historical information about the Sumerian civilization. We generated 100 True/False questions and 100 fill-in-the-blank MCQ questions, with varying numbers of questions for each rule.

Correct Choice	Choice 4	Choice 3	Choice 2	Choice 1	Question
لكش	لكش	إينميباراكيسي	الصين	كيش	ثبتُ الملوك السومري هو وثيقة مسمارية، خطّها كاتب من مدينة______سنة 2100 ق.م
سومر	سومر	أورنمو	كيش	إريدو	ورغم ادعاء كل من الصين والهند بسبق تأسيس المدن، فالأرجح هو أن______هي صاحبة أولى المدن في العالم، ومن أهم مدنها: إريدو وأوروك وأور ولارسا وإيسِن وأداب ولكش ونيبور وكيش وكولا (أو كُلاب)
2100	3696	9138	2100	4287	ثبتُ الملوك السومري هو وثيقة مسمارية، خطّها كاتب من مدينة لكش سنة______ق.م
60	81	60	40	03	أيضًا ابتكر السومريون فعليًا (الوقت)، حين وضعوا نظام العد القائم على الرقم 60، أي أن الدقيقة تتكون من______ثانية، والساعة تتكون من______دقيقة
5000	5000	3292	1062	4842	يُنظر إلى أوروك بوصفها أول مدينة حقيقية في العالم، وقد لاحظ كريمر أن أسماء المدن السومرية لم تكن سومرية بل عبيدية، أي أن العبيديين هُم مَن أسسوا تلك المدن، على الأقل في صورة قُرى، سنة______ق.م

Figure 2: Generated MCQ Questions samples

In Figure 2, a sample of the generated MCQ questions is presented, demonstrating the application of our method to the article, as mentioned earlier. The first column (Right to Left) is the generated question, which is an original sentence from the input text with the selected keyword replaced by a blank. Columns 2-5 represent the selected distractors, and the final column is the correct choice. For the first two generated questions, a proper noun is selected as the keyword, and the distractors are selected from the (NounList), which is filled from the entire input text at the preprocessing step. For the rest of the generated questions, a number is selected as the keywords, and the distractors are Numbers generated randomly. Finally, the replaced keyword is selected as the correct choice for all the questions.

Replaced Number	Replaced Noun	Replaced Adjectives	Rule Name	Answer	Question
61 <- 60			NumberReplacement	خطأ	أيضًا ابتكر السومريون فعليًا (الوقت). حين وضعوا نظام العد القائم على الرقم 61. أي أن الدقيقة تتكون من 61 ثانية. والساعة تتكون من 61 دقيقة صحيح أم خطأ؟
			NoModification	صحيح	بصرف النظر عن أصل العبيديين. فإنهم هجروا حياة الصيد وجمع الثمار إلى حياة الزراعة قبل سنة 5000 ق صحيح أم خطأ؟
	سومر -> أوتوحيكال		NounReplacement	خطأ	كتب أورنمو المدونة القانونية الأولى في أوتوحيكال. لتسبق بذلك قانون حامورابي البابلي الأشهر صحيح أم خطأ؟
	بيرتمان -> شولكي		NounReplacement	خطأ	لاحظ شولكي أن العادة الحالية في تفقد المرء برجه الفلكي ترجع إلى الحضارة السومرية . وأن أول من لاحظ علامات الأبراج التي يولد فيها المرء كان العراقيون القدماء. وإليهم ترجع تسميتها صحيح أم خطأ؟
	سومر -> أور		NounReplacement	خطأ	ورغم ادعاء كل من الصين والهند بسبق تأسيس المدن. فالأرجح هو أن أور هي صاحبة أولى المدن في العالم. ومن أهم مدنها: إريدو وأوروك ولارسا وإيسِن وأداب ولكش ونيبور وكيش وكولا (أو كُلاب) صحيح أم خطأ؟

Figure 3: Generated True/False Questions samples

In Figure 3, a sample of the generated True/False questions is showcased, exemplifying the application of our method to the previously mentioned article. The first column (Right to Left) is the generated question, followed by the correct answer. The third column is the rule name that is used to generate the question. The last three columns represent the replaced Adjective, Noun, or number from the original sentence with the new assigned value. The first question is generated by applying the Number replacement algorithm by selecting a number as the keyword and replacing it with another randomly generated number. The second generated question results from the stochastic element returning false for any applied True/False question generation algorithm so that the generated question is the same original sentence without any modification and the correct answer is True. Finally, the last three generated questions are the result of applying the Noun replacement algorithm, and we can see in the column (Replaced Noun) the original noun and the newly assigned value that is replaced with it in the generated questions.

Since the scientific community uses no standardized method to evaluate the results of AQG, as concluded by the study of [23], and many researchers have used human manual evaluation to evaluate the task of AQG [24, 25], we also used manual evaluation to assess our method for each type of generated question and each rule used.

In our evaluation scoring calculation, we used a simple scoring of one or zero; one indicates that the question is acceptable, and zero indicates that it's not.

For true/false questions, each question is evaluated individually by one or zero, and for MCQ questions, each question and its distractors are evaluated individually by one or zero. The final percentage of acceptable questions is calculated for each type of question.

The evaluation of questions is based on many factors such as clarity, relevance, length and complexity, and Plausibility for the distractors. To provide fairness for our results, four history experts participated in the evaluation process, where each expert evaluated the questions individually, and the average of their combined evaluation was used as our final result.

5. Results and Discussion

Figure 4 illustrates the percentage of acceptable generated fill-in-the-blanks MCQs, standing at 82%. This surpasses the total percentage of True/False questions by 12%, with a rate of 70%.

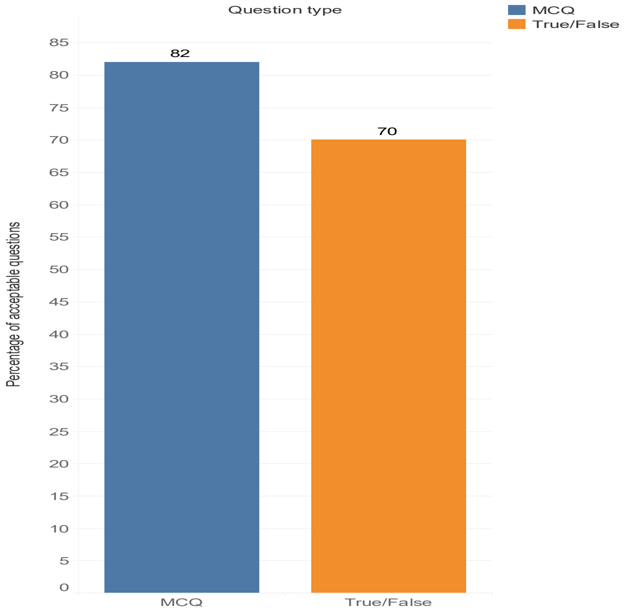

Figure 4: Percentage of acceptable generated questions for each type

Figure 5 delves into further detail for each question type, showcasing the percentage of acceptable questions attributed to each utilized rule.

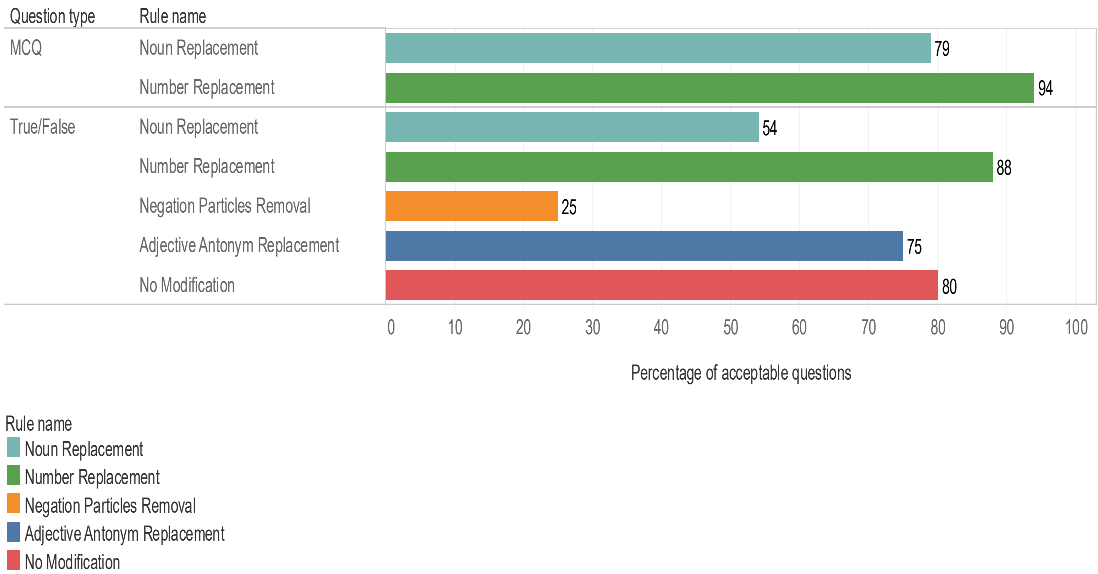

Figure 5: Percentage of acceptable generated questions for each rule

We can observe that the Number replacement rule performed the best for both types of questions by (94%) for the MCQ and (88%) for the True/False questions; this high percentage was because the article used contained a lot of digits rather than written numbers. When we replace a number, specifically digits, in any Arabic sentence, the sentence will remain coherent and grammatically correct. Still, this percentage can drop with the increase of written numbers in the input text, as replacing one written number with another can increase the probability of a grammatical error.

The percentage of the Noun replacement rule in the MCQ questions is (79%), and it performed better than its percentage in the True/False question, which is (54%); this was caused by the fact that when we replace a proper noun from an Arabic sentence with a blank, the sentence will remain structured. However, when we replace a proper noun with another, the sentence can suffer some grammatical issues, and the generated True/False question can become unclear.

The removal of negation particles is the least parentage (25%) in the True/False questions. The complexity of the Arabic language causes this. In some cases, the negation particles are not meant to negate the sentence and removing them will cause the sentence to be poorly structured.

The adjective antonym replacement rule in the True/False questions performed a percentage of (75%), as not all cases of replacing an adjective with its antonym can result in a grammatically sound Arabic question.

And finally (80%) was the percentage of acceptable True/False questions when we didn't perform any modification to the original sentences. From the first impression, one might suggest that when we leave the sentences unchanged, the percentage of the acceptable generated questions should be (100%), but that's not practically true, as in any given text on a specific topic, some sentences rely on the information mentioned in the previous one, so the generated questions may be a sentence taken out of context and unclear as it does not contain any information to generate a True/False question.

One more thing to mention about the results above: they are not fixed and may vary depending on the input text. Many factors will affect the percentage of acceptable generated questions, such as the size of the text, the number of proper nouns, adjectives, negation particles, and the written or digit numbers in the text.

6. Conclusion and Future Scope

The proposed method in this paper can automate or semi-automate the task of AQG for the Arabic language focused on history topics without any reliance on pre-existing knowledge, external sources, or pre-trained models. The method can successfully generate fill-in-the-blanks MCQ questions with distractors and True/False questions along with the correct answers, allowing those generated questions to be automatically graded.

The input text must be relatively large for our method to work efficiently. Ideally, it should contain at least twice the number of sentences needed to generate the desired number of questions. Also, the text should contain an appropriate number of proper nouns, numbers, and adjectives.

Several future tasks can be undertaken to enhance the proposed method and generate other types of questions, such as WH-Questions for the Arabic language. This will require some additional rules to analyze each sentence further and generate the questions with the appropriate distractors along with the correct answer, allowing the functionality of automatic grading for this type of question.

7. Conflict of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. B. Bergbauer, E. A. Hanushek, and L. Woessmann, "Testing," *Journal of Human Resources*, vol. 59, no. 2, pp. 349-388, Mar. 2024.

[2] M. Leenknecht, L. Wijnia, M. Köhlen, L. Fryer, R. Rikers, and S. Loyens, "Formative assessment as practice: The role of students motivation," *Assessment & Evaluation in Higher Education*, vol. 46, no. 2, pp. 236-255, Feb. 2021, doi: 10.1080/02602938.2020.1765228.

[3] E. Godaert, K. Aesaert, J. Voogt, and J. van Braak, "Assessment of students digital competences in primary school: A systematic review," *Education and Information Technologies*, vol. 27, no. 7, pp. 9953-10011, Aug. 2022.

[4] T. Supriyatno, S. Susilawati, and A. Hassan, "E-learning development in improving students critical thinking ability," *Cypriot Journal of Educational Sciences*, vol. 15, no. 5, pp. 1099-1106, Nov. 2020.

[5] W. Wang, T. Hao, and W. Liu, "Automatic question generation for learning evaluation in medicine," in *Advances in Web Based Learning–ICWL 2007: 6th International Conference,* pp. 242-251, Aug. 2008.

[6] B. C. Oguguo, F. A. Nannim, J. J. Agah, C. S. Ugwuanyi, C. U. Ene, and A. C. Nzeadibe, "Effect of learning management system on Students performance in educational measurement and evaluation," *Education and Information Technologies*, vol. 26, pp. 1471-1483, Mar. 2021.

[7] M. Fuadi and A. D. Wibawa, "Automatic Question Generation from Indonesian Texts Using Text-to-Text Transformers," in *2022 International Conference on Electrical and Information Technology (IEIT)*, pp. 84-89, Sept. 2022.

[8] X. Ma, Q. Zhu, Y. Zhou, and X. Li, "Improving question generation with sentence-level semantic matching and answer position inferring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8464-8471, Apr. 2020, doi: 10.1609/aaai.v34i05.6366.

[9] L. A. Tuan, D. Shah, and R. Barzilay, "Capturing greater context for question generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 9065-9072, Apr. 2020, doi: 10.1609/aaai.v34i05.6440.

[10] M. Blšták, "Automatic Question Generation Based on Sentence Structure Analysis," *Information Sciences & Technologies: Bulletin of the ACM Slovakia*, vol. 10, no. 2, Dec. 2018.

[11] E. L. Fasya, "Automatic question generation for virtual humans," University of Twente, 2017.

[12] R. Shah, D. Shah, and L. Kurup, "Automatic question generation for intelligent tutoring systems," in *2017 2nd International Conference on Communication Systems Computing and IT Applications (CSCITA)*, pp. 127-132, Apr. 2017.

[13] D. C. Tsai, A. Y. Huang, O. H. Lu, and S. J. Yang, "Automatic question generation for repeated testing to improve student learning outcome," in *2021 International Conference on Advanced Learning Technologies (ICALT)*, pp. 339-341, Jul. 2021.

[14] K. Mazidi and R. D. Nielsen, "Leveraging multiple views of text for automatic question generation," in *2015 Artificial Intelligence in Education: 17th International Conference,* pp. 257-266, Jun. 2015.

[15] K. Mazidi and P. Tarau, "Automatic question generation: From NLU to NLG," in *2016 Intelligent Tutoring Systems: 13th International Conference*, pp. 23-33, Jun. 2016.

[16] P. Khullar, K. Rachna, M. Hase, and M. Shrivastava, "Automatic question generation using relative pronouns and adverbs," in *Proceedings of ACL 2018 Student Research Workshop*, pp. 153-158, Jul. 2018.

[17] L. Bednarik and L. Kovacs, "Implementation and assessment of the automatic question generation module," in *2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 687-690, Dec. 2012.

[18] B. Diatta, A. Basse, and S. Ouya, "Bilingual ontology-based automatic question generation," in *2019 IEEE Global Engineering Education Conference (EDUCON)*, pp. 679-684, Apr. 2019.

[19] C. Kwankajornkiet, A. Suchato, and P. Punyabukkana, "Automatic multiple-choice question generation from Thai text," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1-6, Jul. 2016.

[20] K. Z. Bousmaha, N. H. Chergui, M. S. A. Mbarek, and L. H. Belguith, "AQG: Arabic Question Generator," *Revue d'Intelligence Artificielle*, vol. 34, no. 6, pp. 721-729, Dec. 2020.

[21] S. A. Yousif, V. W. Samawi, and I. Elkabani, "Arabic text classification: The effect of the awn relations weighting scheme," in *Proceedings of the World Congress on Engineering*, vol. 2, pp. 1-5, Jul. 2017.

[22] A. Belal, "Comprehensive processing for Arabic texts to extract their roots," *Iraqi Journal of Science*, vol. 60, no. 6, pp. 1404–1411, Jun. 2019, doi: 10.24996/ijs.2019.60.6.25.

[23] J. Amidei, P. Piwek, and A. Willis, "Evaluation methodologies in automatic question generation 2013-2018," in *Proceedings of the 11th International Natural Language Generation Conference*, pp. 307–317, Nov. 2018.

[24] C. Van Der Lee, A. Gatt, E. Van Miltenburg, S. Wubben, and E. Krahmer, "Best practices for the human evaluation of automatically generated text," in *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 355-368, Oct. 2019.

[25] C. van der Lee, A. Gatt, E. van Miltenburg, and E. Krahmer, "Human evaluation of automatically generated text: Current trends and best practice guidelines," *Computer Speech & Language*, vol. 67, p. 101151, May 2021.