Journal of Information Systems Engineering and Management
2025, 10(15s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Revolutionizing Education System with Automatic Question Generation Using Rule Based Method

Sanah Sayyed[1*], C Namrata Mahender [2], Ramesh R Naik[3]

[1]Dept of CS and IT, Dr. Babasaheb Ambedkar, Marathwada University Aurangabad, Maharashtra, India. sanahsayyed92@gmail.com

[2]Dept of CS and IT, Dr. Babasaheb Ambedkar, Marathwada University Aurangabad, Maharashtra, India. nam.mah@gmail.com

[3]Dept of CSE, Nirma University, Sarkhej, Ahmedabad. Gujarat. India. rameshram.naik@nirmauni.ac.in

*Corresponding Author:sanahsayyed92@gmail.com

ARTICLE INFO

Received: 07 Dec 2024

Revised: 22 Jan 2025

Accepted: 07 Feb 2025

ABSTRACT

The current research aimed at creating a rule-based method for forming Wh- and Yes/No-type questions based on textual input. The study uses the rule-based method to automatically create Wh- and Yes/No-questions. The approach is based on the syntactic analysis of input sentences to identify the corresponding question forms to be used and apply certain rules for each type. For the proposed method, the achieved accuracy is 82. 20% in generating Wh- and Yes/No-type questions. The findings suggest that the rule-based approach produces appropriate questions corresponding to intervention aims, including checking for understanding and encouraging critical thinking. Such a method may be more effective than many other approaches that are currently used in practice in terms of ease, efficiency, and relevance to education environments. That provides a rich solution that can meet the needs of educators and students alike.

Keywords: Automatic Question Generation,Word Tokenization, POS Tagging, Rule Based Method,Wh type question, Yes/No type question.

INTRODUCTION

Automatic Question Generation (AQG) has become an essential element of the intelligent educational systems that propose the approach to automate the generation of questions for educational testing, learning, and knowledge assessment. The conventional AQG methods have however been advanced over the years to comprise the rule based methods, machine learning approaches, and the integrated techniques. Of these, rule based methods have been preferred since they are easy to understand, easy to implement, and can also include domain specific natural language processing knowledge. Nevertheless, there still are certain challenges for the existing rule-based AQG systems, which prevent them from being more elaborate and efficient. One of the main issues in the rule-based approaches to AQG is a problem of its limited adaptability to various patterns of sentence formation and relatively rigid approach to more complex grammar features. Furthermore, these systems fail to grasp the context and still fail to disambiguation or come up with semantically meaningful questions from complex text.

In order to overcome these shortcomings, this work presents an improved rule-based AQG framework that includes advanced linguistic rules to automatically construct high-quality questions. The proposed method allows for versatility in term of the kind of input text being processed through incorporating both semantic and syntactic analysis.

By using automated systems, large numbers of varied questions can be created quickly. Thus relieving the pressure of the generation of questions from teachers and content writers. Automatic question generation proves to be an essential means, as it automates the process, reduces the load on the teacher, and makes overall evaluation more flexible and individualized. This is not only a time-saving mechanism but also improves the process of teaching and learning [1][2][3].

AQG poses several difficulties, especially when the system tries to generate contextually appropriate and varied questions of high quality. Some of these challenges can be addressed using a rule-based approach whereby question formation is carried out based on some linguistic, semantic, or structural rule set out in advance

LITERATUREREVIEW

Table 1. Literature review

Method	Dataset and Size	Disadvantages
Pretrained Language Model [4]	Custom dataset	Pretrained based models can sometimes fail to capture the details of context of educational material, hence they generate irrelevant questions.
Text-to-Text transformation models [5]	Custom dataset Size of dataset 750	Question generation and answering in the specific domain, which is an important component, has not been

	rows	given adequate attention in the context of current systems.
Rule based method [6]	Custom dataset	The system also fails to properly handle complex syntactic structures and it will therefore sometimes end up generating less desirable or less contextually appropriate questions.
T5 transformer model, and GPT-3 training model.[7]	Real test subjects as a dataset	The research limits the use of advanced models in question generation while ignoring other methods or models that could assist in refining the process.
KNN algorithm [8]	Datasets are from articles	The system still lacks improvements in the areas of context and evaluation methods.
Standalone question generation, visual question generation and conversational question generation.[9]	SQUAD, (23215 document) CMU Q/A Dataset (150 documents)	In this work limited number of conversational datasets is used for automatic question generation.
Rule based method [10]	Squad dataset	Nevertheless, the model can create questions, but the model may lack the background and or more important aspects of the subject area, resulting in improper or inaccurate questions
Gaussian naive base method, rule based method[11]	SQUAD (100000 question generated on Wikipedia article)	Dataset may sometimes choose the wrong word as an answer, pointing out that better training of the system with other domain-specific datasets should be done in order to enhance the levels of intelligence and precision of the system.
Genetic Algorithm[12]	Question bank	The work's drawback is that it creates an ineffective and ambiguous question bank.
RL-based Graph2Seq model[13]	SQuAD. (100000 questions)	The architecture of the model, the graph neural network, and the reinforcement learning might be hard to design and deploy and could need a large amount of computational power.
POS Tagging, Named Entity Recognition, chunking [14]	Educational content	The paper also reveals that in the next tasks the authors intend to enhance the quality of the questions generated, meaning the current questions are not necessarily high quality.
Mind maps [15]	Elementary science learning materials for classes 4, 5, and 6	The principal drawback of the approach used for the question generation is the direct dependency on the quality and coverage of the dataset. In case there are not many samples in the dataset, the questions generated here may not be so helpful.
Rule based method [16]	Custom dataset	The current approach not be using the paragraph level information efficiently and thus, the need to enhance performance particularly for paragraph type questions.

Our proposed system addresses issues as identified in previous literature like is [5],[9],[10] [12]. As employs POS tagging as it checks the correct tags of sentences, and rules are crafted and applied on inputs which generate correct structure and grammatical accurate question bank and result in the generation of accurate and reliable questions. it gives a solid measure of quality for generated questions.

METHODOLOGY

In this study we have tested our proposed system with dataset [17] as we get 92 % accuracy. Dataset contains simple sentence that consists of only one main clause and the simplest of all for a verb to construct. NLP systems work well with simple sentences because there are fewer points of connection between the words, and there are fewer differences in the actual language. With the existing dataset, Because of the small dataset size (only 50 sentences), we were unable to subjectively verify the effectiveness of our suggested approach. To overcome this, we have created our own larger dataset which contains 462 sentences. This new dataset contains more sentences, and it also encompasses a greater variety of scenarios. With the help of this larger dataset, we are able to perform better test and get better results.

The work flow diagram of automatic question generation is explained as below:

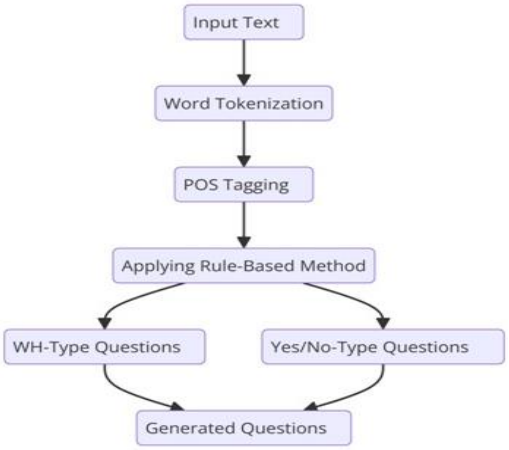

Figure 1.Workflow diagram of proposed system

The proposed system is focused on automatic question formation from dataset of 462 sentences. We first segment the given text into individual words. Then assign them their respective Part of Speech tags. Based on a crafted set of rules, we are then able to form WH-type questions and yes/no types of questions, identifying a particular part of the sentence, namely nouns and verbs. In cases of Yes/No questions, we also recognize auxiliary verbs and place them at the beginning of each question. This leads to the proper development of the questions in the right format and structure. As such, the following steps will allow the development of WH-type as well as Yes/No questions from the input sentences. We have developed rules to generate WH (whose, which, what, when, why, where, who) types of questions and Yes/No types of questions are explained below:

2.1 What type of question

The "what" type of question is commonly used to enquire about the identity, nature, or description of something? To formulate a "what" question, the primary requirement is to determine the subject or object of the sentence that represents the thing being inquired. This can frequently be accomplished by looking for nouns or noun phrases in the statement.

Rules to generate what types of questions,

i)First tokenise the input sentence into words using word tokenize.

ii)Tag each word with its part of speech using a part of speech tagger.

iii) Find a noun ('N' tag) followed by a verb ('VB' tag) in the sentence.

iv) Consider the noun as the subject and the verb as the action.

v) If both a subject and an action are found, construct a "what" type question using them.

Print the generated "what" question for each sentence.

Example:

Input: Earth is known as the 'Blue Planet' due to its abundant water bodies.

Output: What **is** known as the 'Blue Planet' due to its abundant water bodies?

2.2 When type of question:

To form a "when" type of question, look for specific time expressions or temporal adverbs in the sentence. These can include words or phrases that denote a specific point in time, duration, or frequency. Some common examples are:

Specific times: now, tomorrow, yesterday, next week, last month, on Monday, at 3 PM

Duration: three hours, since yesterday, Monday through Friday, all-day

Frequency: daily, weekly, every Monday, once each month.

Rules to generate when types of question

i) Tokenise each sentence into words

ii) Assign part-of-speech (POS) tags to each word.

iii) We initialise an empty list (question words) to store the words in the question.

iv) If a word with a POS tag of 'IN' (preposition) is found, stop the procedure.

v) Construct the "when" question using the words found before the preposition.

vi) Join the words found before the preposition and prepend "when" to the beginning to construct the "when" question.

vii) Print the generated "when" question for each sentence.

Example :

Input : sadaf is going to movie at 9 p.m?

Output: when is sadaf going to movie?

2.3 Why type of question:

To create a "why" type of question in English, you typically look for words or phrases in the sentence that indicate a reason or purpose. These words can includes: Because, Since, As, For, Due to. When you identify these words or phrases in a sentence, you can construct a "why" question by placing "why" at the beginning of the sentence and reordering the rest of the sentence as required.

Rules generate why types of questions

i) Tokenise sentence into words

ii) Assign part-of-speech (POS) tags to each word.

iii) We initialise an empty list (question_words) to store the words in the question.

iv) If a word with a POS tag of 'IN' (preposition) is found, it stops the procedure.

v) Constructs the "why" question using the words found before the preposition.

vi) Joins the words found before the preposition with spaces and prepends "why" to the beginning to construct the "why" question.

vii) Prints the generated "why" question for each sentence.

Example :

Input : They launched a social media campaign because of its effectiveness in reaching a wide audience.

Output : Why they launched a social media campaign ?

2.4 Who type of question:

A "who" type of question is used to identify a person or people performing an action by identifying the subject using proper nouns or pronouns. The question is formed by placing "who" at the beginning of the sentence.

Rules generate who types of questions

i) Tokenise sentence into words

ii) Apply POS tagger to assign the tags to words.

iii) Identify the subject (in this case, the name of a person) by finding the first word in the sentence that is either a proper noun (NNP) or an adverb (RB).

iv) Generates the "who" question by removing the identified subject from the sentence and appending a question mark.

v) Print the generated "who" question for each sentence.

Example :

Input : Emily went for a picnic in the park.

Output: Who went for a picnic in the park?

2.5 Where type of question:

To create a "where" question, the main condition is to identify the location or place described in the sentence. This can often be done by looking for words that indicate a location, such as prepositions like "in," "on," "at," "by," etc., or specific place names.

Rules generate where types of questions

i) Tokenise sentence into words

ii) Assign part-of-speech (POS) tags to each word.

iii) If a word with a POS tag of 'IN' (preposition) is found, stop procedures.

iv) Constructs the "where" question using the words found before the preposition.

v) Joins the words found before the preposition and prepends "where" to the beginning to construct the "where" question.

vi) Print the generated "where" question for each sentence.

Example :

Input : Noor placed the book on the shelf.

Output: where Noor placed the book ?

2.6 Which type of questions:

In order to form a “which” type of question, one has to look for a particular choice or an option stated in the sentence. This kind of question is usually intended in a way that seeks to establish which option is desirable or which option needs to be singled out from the rest of the options.

Rules generate which types of questions

i) Split the string of the input sentence into words using word tokenization.

ii) Label each term with its part of speech using the pos of tag. Identify an object in the sentence that was tagged by ‘N’ followed by a verb tagged by ‘VB’. The noun is used as the object, while the verb is used as the process. If both an object and process are detected, build a “Which” type question using it. Print the generated "which" question for each sentence.

Example :

Input : The Mona Lisa is known as the 'Masterpiece of Leonardo da Vinci' due to its artistic value.

Output: Which is known as the 'Masterpiece of Leonardo da Vinci' due to its artistic value?

2.7 Whose type of questions:

The major requirement for constructing a “whose” type of question is to pick the feature involving possession/ownership in the given sentence. This involves recognising the contract possessive nouns or pronouns, which show that an item belongs to someone or something.

Rules generate whose type of questions

i) Sentence is broken into words using word tokenization.

ii) Each word in the input is assigned its tags by using POS tagging. This is accomplished by the POS-tagged words and finding those that were tagged as possessive endings (POS).

iii) If a possessive ending is identified, it is presupposed that the noun preceding the possessive ending functions as the possessive noun in cases where the noun is labelled as 'NN' (common noun) or 'NNP' (proper noun).

iv) The sentence is transformed into a question. The question mark is added.

Example :

Input : Ramya takes Neela's address.

Output: Ramya takes Whose address.?

2.8 Yes/No type of questions

The basic requirement for making a question of the "yes/no" type is to select a statement that can be affirmed or negated. These questions involve simply changing a statement into a question by using an auxiliary verb, modal verb, or phrase, where the expected response is 'yes' or 'no'.

Rules generate yes/no type of questions

i) Split the given input string into a list of words using word tokenization.

ii) Provide a list of modal/auxiliary verbs.

iii) Find First Auxiliary/Modal Verb Location Identify the first auxiliary or modal verb in the tokenised sentence.

iv) Generates a yes/no question, If we get a modal/auxiliary verb, take out the modal/auxiliary verb and capitalise it.

v) Rewrite the given sentence, inserting the auxiliary/modal verb at the beginning of the sentence and placing the rest of the words at the end.

vi) The reconstructed sentence is returned as a question. Create the yes/no question from the given sentence, then print the generated question.

Example:

Input: Sophie will be singing in the concert.

Output: Will Sophie be singing in the concert?

RESULTS AND DISCUSSION

The proposed system is generating automatically questions from the given corpus of 462 sentences. Our first step is to split the text into words and assign each word the POS tag that belongs to it. This makes it possible to determine crucial components like nouns and verbs important in developing WH-type questions ("What," "who," "when," "where," "which," "why"," whose"). In the case of yes/no questions, the system identifies the auxiliary verbs and moves them right to the front of the proposed question structure. This method helps in making sure that the questions are in correctly structured grammar and in the right context. Based on the rule-based approach used in generating these questions, the results have been impressive, with an accuracy of 82.2%. Therefore, the high degree of correctness of the produced questions proves the efficiency of the system in terms of generating syntactically and semantically correct questions. Table 2 shows the result of automatic questions generated; Figure 2 shows the graphical representation of produced questions. The accuracy of given automatic questions is measured by the following equation (1)

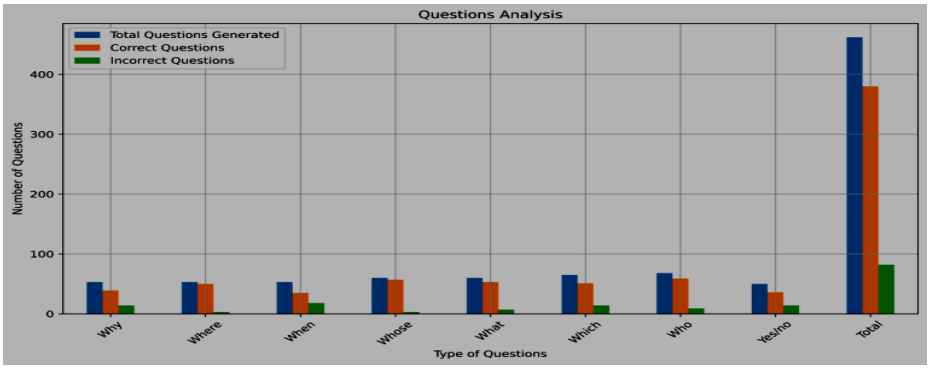

Figure 2. Result of the total no of questions are generated in graphical form

Table 2. Result of Total Questions Generated

Type of Questions	Total Questions Generated	Correct Questions	Incorrect Questions	Accuracy (%)
Why	53	39	14	73.58
Where	53	50	3	94.34
When	53	35	18	66.04
Whose	60	57	3	95
What	60	53	7	88.33
Which	65	51	14	78.46
Who	68	59	9	86.76

Yes/no	50	36	14	72
Total	**462**	**380**	**82**	**82.25**

$$\text{Accuracy} = \frac{\text{Total no of correct generated questions}}{\text{Total no of generated questions}} * 100 \qquad \text{Equation (1)}$$

From the above results, as shown in Table 2 it is suggested that "Whose" types of question have highest accuracy that is 95% accuracy and the second most accurate are "Where" questions that is 94.34%. There are less accurate questions are "When" questions, with 66.04% accuracy. The when type of questions have lower accuracy because of possible problems with the identification of temporal features in the text. It seems that to enhance the accuracy of "When" questions, there are several useful things to do. First, by using more examples that are related to dates, time and events related information additional to the training data set, the system can detect the pattern more proficiently. We can also make improvements by concentrating on words and phrases which indicates time or we can formulate rules to be followed by the system to make it understand phrases of time. These changes can help for getting a correct generation of "When" questions. In order to assess the effectiveness of the proposed system, the obtained results compared to previously published research work which adopted rule-based approach as shown in Table 3. Our proposed system, also applies rule based method, has shown a comparatively higher accuracy of 82.20%. This considerable improvement indicates the significance of the changes made by our approach. The contribution of our work is primarily on enhancement of the rule-based mechanism along with richer linguistic rules for "Wh and Yes/No" question generation. These modifications are to overcome the problems identified in previous studies like [6][10][16], the coverage of rules and accuracy issue. Apart from providing better accuracy for the rule-based question generation systems, our study also fills in the gaps of the prior methodologies and offers a strong framework for increasing the usability of such systems in practice.

Table 3. Comparision of previous work with propsed work

Method	Dataset	Types of question	Result
Rule based method [6]	Custom dataset	Wh type of questions	Blue score:0.718,F1:0.5
Rule based method[9]	Squad dataset	Fill in the blanks (FIB) and wh type of questions	Success rate of FIB :59 Success rate of wh type of questions : 49
Rule based method [10]	Squad dataset	Multiple choice question and fill in the blanks questions	Not specified
Rule based method [16]	Custom dataset	Deep questions	Accuracy score 3.60
Proposed System			
Method	Dataset	Types of question	Result
Rule based method	Custom dataset	Wh type and Yes/No type	Accuracy: 82.20%

CONCLUSION

Automated question generation (AQG) is a process of creating questions from text content with the help of natural language processing. AQG can save time in generating assessments, ease the creation of conversational agents, and help find and verify knowledge effectively in scientific research. The proposed method efficiently generates the WH-type and Yes/No questions in the dataset using rule-based approaches. Based on text segmentation and POS tagging, the development and structure of questions are maintained properly according to a set of heuristics. The achieved accuracy of 82.20 % represents a good result. There are several challenges that need to be addressed in future work regarding automatic question generation: there are issues with the negation, context dependency, etc.

REFERENCES

[1] Lili Wang, Ruiyuan Song, Weitong Guo & Hongwu Yang ,Exploring prompt pattern for generative artificial intelligence in automatic question generation, Interactive Learning Environments, 2024.DOI: 10.1080/10494820.2024.2412082

[2] Manisha Mali , Rutvik Kachgunde, Automated Question Generation and Grading Using Artificial Intelligence, International Journal for Multidisciplinary Research, 2024. Available from: https://www.ijfmr.com/papers/2024/6/30345.pdf

[3] Velmurugan, H., Pandiarajan, N., Ravi, L., Voleti, S., Sheik, R., Nirmala, S.J. Automatic Question

Generation:A Comparative Analysis of Rule-Based and Neural Network-Based Models. In: Mahapatra, R.P., Peddoju, S.K., Roy, S.,Parwekar, P. (eds) Proceedings of International Conference on Recent Trends in Computing. ICRTC Lecture Notes inNetworks and Systems, vol 954. 2024. Available from: https://doi.org/10.1007/978-981-97-1724-8_21

[4] Dong, Shi., Shunyi, Zhao., Qingting, Liu., Yi, Zhang, Research and Design of Automatic Questioning System Based onQuestion Generation. 2024. Doi: 10.1109/iciet60671.2024.10542827

[5] Sahil Kale , Gautam Khaire , Jay Patankar, FAQ-Gen: An automated system to generate domain-specific FAQs to aidcontent comprehension. Journal of Computer-Assisted Linguistic Research, 2024. Available from:https://doi.org/10.48550/arXiv.2402.05812

[6] Walelign Tewabe Sewunetie, Rule-Based Automatic Question Generation using Dependency Parsing, research square,2023. Available from: doi: 10.21203/rs.3.rs-2693893/v1

[7] Docca Pranav , Badri Prasad V R .Histobot: Question Generation System Using Deep Learning Techniques.International Conference on intelligent computing technologies and Research (i-COMPUTER), 2023. Availablefrom:https://doi.org/10.47392/irjash.2023.S071

[8] Lala Septem Riza, Yahya Firdaus, Rosa Ariani Sukamto, Wahyudin, Khyrina Airin Fariza Abu Samah, .Automaticgeneration of shortanswer questions in reading comprehension using NLP and KNN. Multimedia Tools and Applications,2023. Available from: DOI:10.1007/s11042-023-15191-6

[9] Nikahat Mulla, Prachi Gharpure, Automatic question generation: a review of methodologies, datasets, evaluationmetrics, and applications. Progress in Artificial Intelligence 12:1–32. 2023 Available from: DOI:10.1007/s13748-023-00295-9

[10] P. R. Rao, T. N. Jhawar, Y. A. Kachave and V. Hirlekar, Generating QA from Rule-based Algorithms, International 9/9Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, DOI: 10.1109/ICEARS53579.2022.9751723.

[11] Parth Panchal, Janak Thakkar, Veerapathiramoorthy Pillai, Prof. Shweta Patil, Automatic Question Generation andEvaluation. Journal of University of Shanghai for Science and Technology,2021. Available from: DOI:10.51201/JUSST/21/05203

[12] Dimple Valayil Paul ,Automatic Question Paper Pattern Generation using GA Approach, INFOCOMP, v. 19, no. 1, p.2020. Available from: https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/774

[13] Yu Chen, Lingfei Wu, Mohammed J. Zaki, Reinforcement Learning Based Graph-To-Sequence Model For NaturalQuestion Generation. ICLR 2020.Available from:https://doi.org/10.48550/arXiv.1908.04942

[14] Priti Gumaste, Shreya Joshi, Srushtee Khadpekar, Shubhangi Mali. Automated Question Generator System Using NLPLibraries. International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 06 June 2020.Available from:https://www.irjet.net/archives/V7/i6/IRJET-V7I6848.pdf

[15] Selvia Ferdiana Kusuma , Daniel Oranova Siahaan, Chastine Fatichah, Mohammad Farid Naufal. Automatic QuestionGeneration with Classification Based On Mind Map. International Joint Conference on Computer Science and SoftwareEngineering (JCSSE), 2019. Available from: DOI: 10.1109/JCSSE.2019.8864208

[16] Onur Keklik, Rule-Based Automatic Question Generation Using Semantic Role Labeling, IEICE Trans. Inf. & Syst.,Vol.E102–D, No.7 2019. Availablefrom:https://www.jstage.jst.go.jp/article/transinf/E102.D/7/E102.D_2018EDP7199/_pdf/-char/en

[17] Dataset Available from:https://github.com/NishantTanwar/Question-Generation/blob/master/input.txt