PolyAQG Framework: Auto-generating assessment questions

Tee Hean Tan
Center for American Education
Sunway University, Sunway City
teeheant@sunway.edu.my

Phoey Lee Teh
Department of Computing and Information System
School of Engineering and Technology
Sunway University, Sunway City
phoeyleet@sunway.edu.my

Zaharin Yusoff
Department of Computing and Information System
School of Engineering and Technology
Sunway University, Sunway City
zarinby@gmail.com

2021 IEEE International Conference on Computing (ICOCO) | 978-1-6654-3689-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOCO53166.2021.9673579

Abstract— Designing and setting assessment questions for examinations is always a necessary task for educators. In this article, we identify the research gaps in (semi-) automatically generating questions by evaluating all the available approaches developed thus far. We then propose a framework that puts together previous approaches and suggests ways to fill in their gaps. One hundred and thirteen pieces of literature relevant to question generation approaches have been reviewed and compared. For each of the approaches, the uniqueness of the techniques is explained. The PolyAQG Framework is presented with an explanation of how it would contribute to the solution of the problem, by improving the variety of the questions, increasing the total number of possible choices of question selections, as well as providing a better quality of questions. Apart from the framework, another novelty in this work is the innovative way a domain ontology is used to generate a wider variety of questions.

Keywords—knowledge base, ontology, natural language processing, linguistic structures

I. Introduction

Automated question generation (AQG) from text is one of the popular methods to aid educators in preparing assessment questions [1]. With AQG, educators can reduce the cost and effort in preparing the questions and spend more time on other critical instructional activities [2]. Template-based, Syntax-based, Semantic-based, and Ontology-based question generations are the four common approaches used by AQG researchers. The template-based approach uses the pre-defined question templates with empty placeholders filled from the input text [3]. The Syntax-based approach leverages the syntactic features of the input sentence that applies techniques to convert input sentences into questions [2]. To generate Semantic-based questions, one must understand the input sentence's semantic meaning, over and above the sentence's lexical and syntactic structure. More innovatively, the Ontology-based questions will replace relevant words from the domain ontology knowledge-base while constructing the questions.

Each of these approaches has its advantages and limitations. The question formats for the Template-based questions are limited because the question templates define the surface structure of the questions using fixed texts [2]. Syntax-based questions are generated from the input sentence, which uses most of the words from the input sentence to form the question sentence, making the question very similar to the input sentence [2]. The Semantic-based approach provides a deeper level of analysis of the input sentence than the Syntactic-based approach. However, it does not guarantee the quality of the questions, especially on the syntax and the proper grammar used for the generated questions [2]. So far, most of the Ontology-based questions are multiple-choice questions that use the knowledge base to generate a correct answer (key) and the wrong answers (distractors) [4][5][6].

To overcome the shortcomings and to leverage the features of each of the mentioned QG (Question Generation) approaches, we propose a new end-to-end PolyAQG framework, applying all the above four generic approaches, with each of these approaches are replaceable with others. The number of generated questions will increase from one phase to another phase when more information is extracted from the input sentence. This will improve the variety of the questions, increase the total number of possible questions, and provide a better quality of questions. All the generated questions will be saved into a centralized database, so that, the subject lecturers can choose the relevant questions from the database while preparing the assessment paper.

It is clear that the targeted users of an implemented PolyAQG system will be educators, be it teachers, lecturers, examination syndicates, etc.

II. Related Work

Four basic approaches were designed many years back for questions generation; namely, 1) Template-based approach [2], 2) Syntactic approach [2], 3) Semantic approach [2], and lastly, Ontology-Based approach [7]. Each of which has its aspect of focus and prevalent choice of features. Following this, we compare their uniqueness and their differences.

Question Generation was pioneered in 1976 by John H. Wolfe [8], who developed an auto-generated question prototype to assist the independent study of technical manuals and books. The program compares a selected text sentence against a table of pre-stored patterns. If the sentence fits a pattern, a certain kind of question is generated from it. Seyler [9] also used the Template-based approach in his AQG. He developed multiple-choice questions by selecting a named entity from the knowledge graph as its answer. He added a structured triple-pattern query upgrade with the template-based method to convert the query into questions. This method organizes and aligns with the traditional pen and paper-based approach, but making it automated on a system allows multidirectional queries more efficient.

Unlike the Template-based approach, Mitkov and Ha [10] used a simple set of transformational rules (a syntactic approach), a shallow parser, etc., to generate multiple-choice test questions and distractors, while Manish Agarwal [11] developed gap-fill questions with another kind of the syntactic approach. The uniqueness of this latter approach is that it blanks all the keys from the informative sentence, identifies the standard discourse connectives on the sentence to determine the question type, and then applies syntactic transformations to generate the questions.

Following that, around 2009, Chen, Aist, and Mostow [12] generated self-questioning instructions focusing on children's text. They developed questions from both narrative and informational text. The narrative text involves characters, their behavior, and the mental states that drive it. The informative text generally emphasizes descriptions and explanations, often used to introduce an objective phenomenon, as in a sentence. Informational text is an essential source of knowledge, and the young reader can benefit from it. On another level, Fattoh, Aboutabl, and Haggag [13] used Semantic Role Labeling and Named Entity Recognition as a preprocessing step to convert the input sentence into a semantic pattern. This method classified the question type patterns and found the best matching pattern to generate the WH-questions like who, when, where, why, and how. This method increases the variety of choices in question generation. For instance, a statement can generate different questions based on the words used in the answer - such as, to create a question, a noun could be referred to and manipulated to match with the Wh "who", and words that refer to a location would be compared with Wh "where" etc.

On a more advanced level, around the year 2008, Papasalouros, Kanaris, and Kotis [14] implemented the first prototype to generate multiple-choice questions (MCQ) from ontological axioms and asserted/inferred knowledge of a knowledge base developed in Ontology Web Language (OWL). They defined eleven strategies to generate correct answers and distractors. They found out that property-based strategies may produce a more significant number of multiple-choice questions, but they are challenging to manipulate syntactically. Class and terminology-based strategies are much easier to handle syntactically, but they generate fewer questions than ontologies of the same depth and population. In 2013, Chen, Ran, and Zhu [15] built a domain knowledge ontology with a complete description of different concepts and their relationships to generate ontology-based multiple-choice questions. They have identified knowledge relationships in the ontology, such as hierarchy relationships, belongs-to, binary-associations, constraints, etc. While developing the distractors for the multiple-choices questions, they ensured that the word property of distractors and the correct answer must stay the same. Both must have incidence relation in meanings

To explore and understand all the different (analytical) approaches used in generating questions in the past 20 years. We had collected 113 papers from 10 various sources, all of which are relevant to the topic of discussion, including references from IEEE (Institute of Electrical and Electronics Engineers), ACL (Association for Computational Linguistics), Scientific.net, arxiv.org, etc. using the keywords "Automated question generation", "AQG", "template", "Syntactic", "semantic", "ontology", "generate questions" and more.

From the 113 article collection of literature review, we recognized that most of the researchers used one or two approaches in their automated question generation system development, which somehow resulted in the limited choice of question varieties, question types, and the number of the generated questions. This could reduce the flexibility of the academician if they were to utilize and focus on each selection of approach.

We have tabulated the collection and categorized it based on the type of question generated. This allowed us to understand the most prevalent methods used for each selection of approaches and provided us with a bird's eye view of all the approaches weighing by the percentage of literature in each approach. Based on the AQG approach and question type, Table I compares the ratio of papers from our 113 collections.

TABLE I. SHOWS THE AQG APPROACHES AND THE GENERATED QUESTION TYPE

AQG Approach	**No of papers**	**Question type**			
		Gap-fill	***Multiple-choice***	***WH Short-answer***	***Undefined***
Template-based	23%	**84%**	X	11%	5%
Syntactic-based	32%	23%	25%	70%	7%
Semantic-based	13%	10%	25%	62%	3%
Syntactic and semantic-based	6%	X	50%	50%	X
Ontology-based	20%	2%	78%	20%	0%
Others	6%	2%	X	78%	X

From table I, we notice that template-based AQG has dominated the Gap-fill method in its approach, and for syntactic-based semantic-based, both are more prevalent in the use of Wh short answer. Ontology is more towards the multiple-choice direction.

We now explain in detail how the above restricts the flexibility of the question generation. For instance, it is challenging to build question templates for generic topics; thus, a Template-based approach is recommended for special-purpose applications [16]. Writing question templates takes time, where the instructor must carefully craft the Templates to judge the question's difficulty level and ensure that each question's distractors are meaningful [17]. Syntactic-based questions are generated from the input sentence; once the test taker is familiar with the input text, it is easy to answer them. However, syntax-based questions are recommended for short-answer questions and gap-fill questions [2]. Hence, the selection of using this method can allow only short-answer questions to be constructed. Short-answer questions are questions built by selecting an appropriate wh-word and then rearranging the words from the chosen sentence to form short-text answers. The gap-fill, also known as fill-in-the-blanks problems, is made by removing a word or phrase from a text segment [2]. The Semantic-based approach provides a deeper level of analysis on the input sentence than the Syntactic-based method. However, this approach does not guarantee the quality of the questions, especially for the syntax and the proper grammar used on the generated question [2]

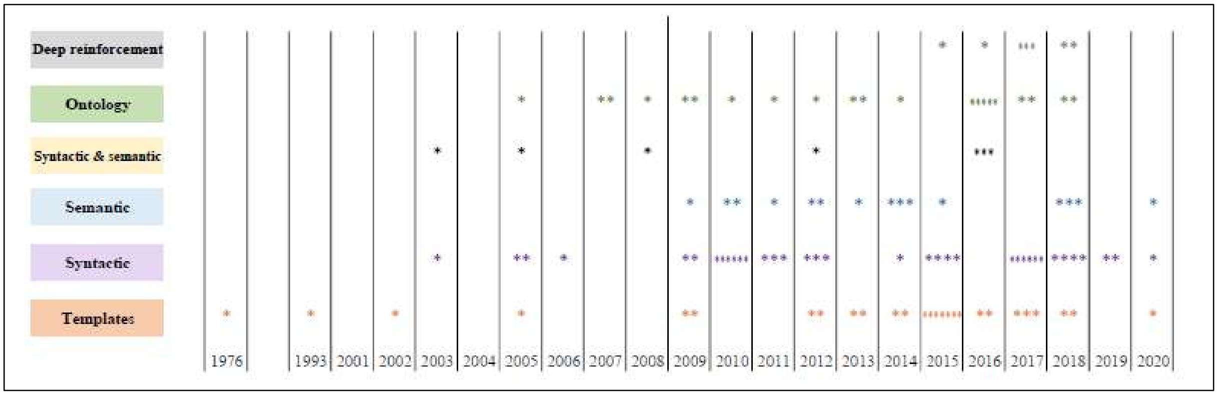

Fig. 1. Timelines for the Different Approaches

We have also drawn a timeline on the different approaches from our collection, as given in Fig.1. The pioneering AQG paper was published in 1976 using the Template-based approach, followed by one in 1993, about eight years later. From 2009 to 2018, there was a tremendous increase in publications on AQG using different techniques within these ten years.

There is also a more recent approach for AQG termed deep reinforcement [18]. The approach is more empirical than analytical, thus quite fundamentally different from the earlier four approaches. It can be quite readily seen that the four analytical approaches are very much related to each other, being on the well-known spectrum of linguistic structures, namely:

- surface/words/morphology (template)
- syntax
- semantics/meaning
- knowledge (ontology).

It should thus be possible to put these four approaches together into a single framework to leverage each other, which is precisely the proposal in this paper.

III. Framework

We propose an end-to-end PolyAQG framework as shown in Fig. 2; a semi-automated structure with reusable and incremental modules. It comprises a fair percentage of the basis of each of the earlier approaches along with their generation methods. We define different levels, or a sequence, of sentence linguistic structures to serve as the base to generate the questions.

Besides analyzing for the linguistic structures of the input sentence and generating the questions from each structure, counting the number of possible questions that can be generated from the input sentence at each linguistic structure level can be done using mathematical permutation, combination, etc., but this will be reported elsewhere. .

There is a total of 10 phases in the Framework. It starts with the Upload module (phase 1.0) by having the user load a relevant source document for questions generation. The Extract sentences module (phase 2.0) will read and extract complete sentences from the uploaded file. The extracted sentences will then go through a series of NLP (Natural Language Processing) transformations in the Construct Morpho-syntactic Structure module (phase 3.0), namely essentially morpho-syntactic parsing, which is made up of two parts:

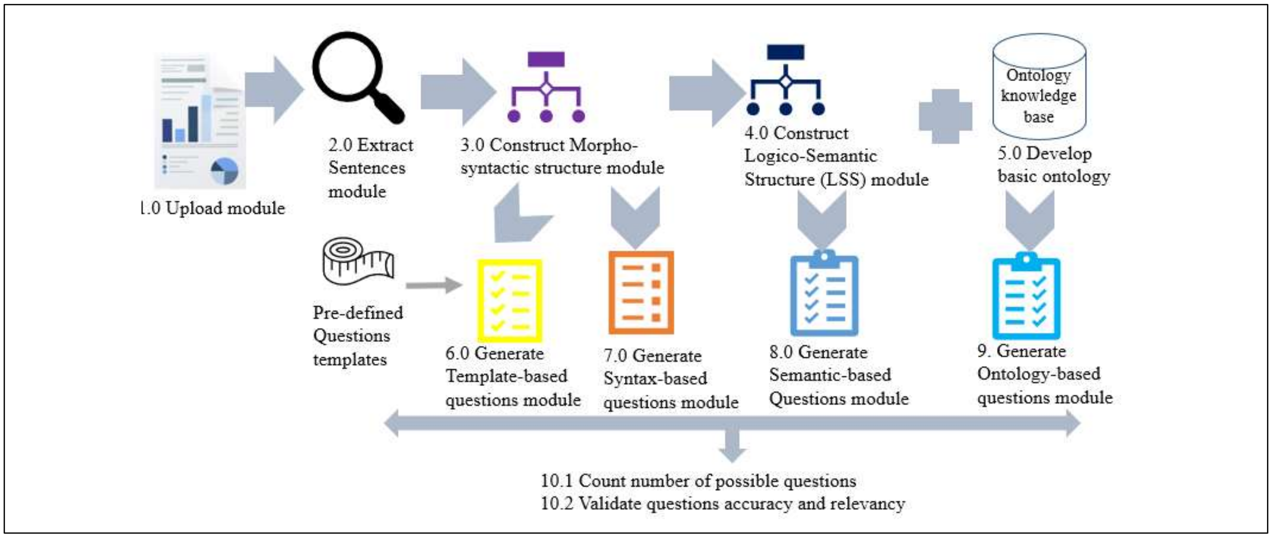
1.0 Upload module
2.0 Extract Sentences module
3.0 Construct Morpho-syntactic structure module
4.0 Construct Logico-Semantic Structure (LSS) module
Ontology knowledge base
5.0 Develop basic ontology
Pre-defined Questions templates
6.0 Generate Template-based questions module
7.0 Generate Syntax-based questions module
8.0 Generate Semantic-based Questions module
9. Generate Ontology-based questions module
10.1 Count number of possible questions
10.2 Validate questions accuracy and relevancy

Fig. 2. PolyAQG Framework

- Morphological analysis – consisting of tokenization, lemmatization, and proper nouns tagging, resulting in a sequence (or pattern) of lemmas with their Part Of Speech (POS) tags, sometimes referred to as the POS structure.
- Syntactic analysis – resulting in a tree of phrase structures and their syntagmatic categories (S, VP, NP, and etc.) as well as syntactic functions (GOV, SUBJ, OBJ, and etc.). This tree is referred to as the morpho-syntactic structure.

Any of the ready-made NLP parsers, such as the Stanford NLP parser [19], should be able to perform these transformations. .

Following this, the Construct Logico-Semantic Structure (LSS) module (phase 4.0) takes the morpho-syntactic structure to form the Logico-semantic Structure, often referred to as the level of meaning. It is a dependency structure formed from the predicates and their arguments within the input sentence. Each node will have their semantic features (PROC, HUM, LOC, and etc.) and each arc is labelled with the logico-semantic relations/roles between the nodes (ARG0, ARG1, DEST, and etc.). An example is given in Fig. 3. It is from the above structures that the questions are generated.

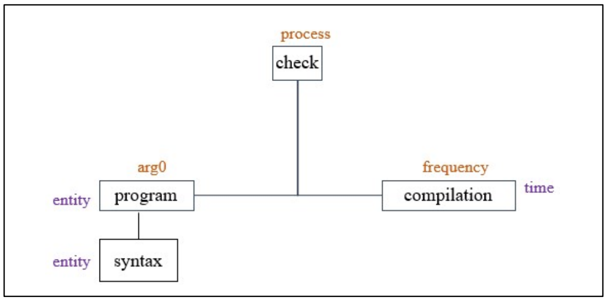

Fig. 3. A sentence in logico-semantic structure

The generate Template-based question module (Phase 6.0) starts by taking the POS structure of the extracted sentence to match against the available question templates. If a matching is found, this phase will continue with generating the template-based questions. Three question formats are developed in this phase: True-false, Gap-fill, and the WH short-answers questions. The True-false questions are generated through verb-fronting by moving the available verb found from the POS list of the input sentence to the beginning of the sentence to convert the input. Gap-fill questions are generated by systematically leaving out the nouns or verbs in the sentence to leave gaps. Thirdly, from the proper nouns of the sentence, WH short-answer questions are generated. If the proper noun is a person, then a WH-Who question will be generated; otherwise, a WH-What question will be generated.

Continuing to next phase, the generate Syntax-Based question module (phase 7.0) will take the morpho-syntactic structure and its internal information to form new sentences, essentially through structural transformations based on the syntactic functions. Examples include active to passive (or vice-versa), and reordering of objects and complements. The resulting sentences can then be sent back to phase 6.0 to match against the question templates to develop a new set of questions.

One additional set of syntactic-based questions is generated by decomposing the main verb in the sentence [20]. An auxiliary verb is used with a main verb to help express the main verb's tense, mood, or voice. The primary auxiliary verbs are ‘to be’, ‘to have’, and ‘to do’. If an auxiliary verb is not present, the main verb will be converted into its base form. The new sentence will then be restored to its opposite voice sentence, either to passive or active voice, then matched against the question templates in phase 6.0 to develop another new set of questions. Thus, this will increase the total number of the generated questions.

It is not possible to generate the WH-where or WH-when questions from the POS structure nor from the morpho-syntactic structure because these structures do not have semantic features. However, the logico-semantic structure (phase 4.0) has the necessary semantic features, and so such questions on WH-where (from LOC, location), WH-whom (from HUM, human), and etc., can be generated. Furthermore, the additional information on logico-semantic relations would also enable further questions, such as for From-Where (SOURCE), Where-To (DESTINATION), Why (CAUSE), and etc. This is done by the Generate Semantic-based questions module (phase 8.0), which generate questions based on the additional information of semantic features and the logico-semantic relations (typically the said WH-questions).

An ontology is a formal description of knowledge as a set of concepts within a domain and the relationships that hold amongst them. Each concept also has attributes and their values. In the Develop basic ontology module (phase 5.0), an ontology of a specified domain is set up based on several primary relationships: vertical, lateral, and some others. Vertical relations are the inheritance and aggregation (part-of) relations, while lateral relations include synonymy and antonymy, whereas others include ‘belongs to’ or ‘owner of’ relations. Phase 5.0 is a very necessary step to set up a base ontology domain specific ontology using concepts, attributes and various relationships.

The Generate Ontology-Based question module (phase 9.0) will match concepts from the logico-semantic structure against the ontology, and extend and/or replace them via the said primary relations to create new sentences. With the new format of the sentences, they are to compare against the question templates in phase 6.0 to create more questions. This way of using relations within the ontology can be considered novel compared to earlier approaches.

Phases 5.0, 8.0 and 9.0 are undoubtedly highly complex compared to phases 6.0 and 7.0, and are beyond the scope of this paper, and will be reported elsewhere quite soon [19]. It is also quite possible to count the number of possible questions in each of the phases 6.0. 7.0, 8.0, 9.0 with the use of various methods of permutations and combinations, but again this is beyond the scope of this paper.

All the generated questions are compiled and counted in the last phase, the Count number of possible questions module (phase 10.1). These questions are then evaluated manually by the domain experts in the final phase (phase 10.2) in terms of the accuracy and relevancy, as well as the grammar used, ambiguity of words, question structure or format, and question relevancy in terms of the level of knowledge required (broom taxonomy) and questions coverage. Only the correct and relevant questions are chosen to be used to generate the assessment paper. Unused questions are kept in the database for future usage or reference.

IV Implementation

A prototype system for the Framework is under development on the Java platform. The prototype will use the Stanford Parser API to perform the fundamental NLP transformations. It is also connected to the MySQL database to store the input sentences, the extracted sentences, the questions templates, the domain ontology knowledge, and the generated questions. Much more design and development work will be carried out here to increase the level of efficiency and sophistication. In the meantime, quite a number of the highly complex phases are currently being performed manually.

V CONCLUSION

In this paper, an end-to-end question generation framework, PolyAQL is presented. The Framework combines the four generic question generation approaches: the template-based approach, syntactic-based approach, semantic-based approach, an ontology-based approach to generate questions in order to increase the number of the generated questions, improve the variety of the questions, and provide a better quality of the questions. Other than the framework, the innovative way of using relations in a domain ontology to generate a wider variety of questions can be considered as novelty in this work.

References

[1] M. Heilman, V. Aleven, W. W. Cohen, D. J. Litman, and N. A. Smith, “Automatic Factual Question Generation from Text Thesis Committee,” *Diss. Carnegie Mellon Univ.*, 2011, [Online]. Available: www.lti.cs.cmu.edu.

[2] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, “A Systematic Review of Automatic Question Generation for Educational Purposes,” *Int. J. Artif. Intell. Educ.*, vol. 30, no. 1, pp. 121–204, 2020, doi: 10.1007/s40593-019-00186-y.

[3] E. Sneiders, “Automated question answering using question templates that cover the conceptual model of the database,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2553, pp. 235–239, 2002, doi: 10.1007/3-540-36271-1_24.

[4] M. Al-Yahya, “OntoQue: A question generation engine for educational assesment based on domain ontologies,” *Proc. 2011 11th IEEE Int. Conf. Adv. Learn. Technol. ICALT 2011*, pp. 393–395, 2011, doi: 10.1109/ICALT.2011.124.

[5] V. E., T. Alsubait, and P. S. Kumar, “Modeling of Item-Difficulty for Ontology-based MCQs,” 2016, [Online]. Available: http://arxiv.org/abs/1607.00869.

[6] K. Stasaski and M. A. Hearst, “Multiple Choice Question Generation Utilizing An Ontology,” no. 2011, pp. 303–312, 2018, doi: 10.18653/v1/w17-5034.

[7] T. Alsubait, B. Parsia, and U. Sattler, “Automatic generation of analogy questions for student assessment: An ontology-based approach,” *Res. Learn. Technol.*, vol. 20, no. SUPPL, pp. 95–101, 2012, doi: 10.3402/rlt.v20i0.19198.

[8] J. H. Wolfe, “An Aid to Independent Study through Automatic Generation (Autoquest),” *SIGCSE ’76 Proc. ACM SIGCSE-SIGCUE Tech. Symp. Comput. Sci. Educ.*, p. Pages 104–112, 1975, [Online]. Available: https://doi.org/10.1145/800107.803459.

[9] D. Seyler, “Knowledge Questions from Knowledge Graphs,” *ICTIR ’17 Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, p. Pages 11–18, 2017, doi: 10.1145/3121050.3121073.

[10] R. Mitkov and L. A. Ha, “Computer-aided generation of multiple-choice tests,” in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing* -, 2003, vol. 2, pp. 17–22, doi: 10.3115/1118894.1118897.

[11] M. Agarwal, R. Shah, and P. Mannem, “Automatic question generation using discourse cues,” *ACL Work.*, no. June, pp. 1–9, 2011.

[12] W. Chen, G. Aist, and J. Mostow, “Generating Questions Automatically from Informational Text,” *Work. Proc. AIED 2009 14th Int. Conf. Artif. Intell. Educ.*, no. 1, pp. 17–24, 2009.

[13] I. E. Fattoh, A. E. Aboutabl, and M. H. Haggag, “Semantic Question Generation Using Artificial Immunity,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 7, no. 1, pp. 1–8, 2015, doi: 10.5815/ijmecs.2015.01.01.

[14] A. Papasalouros, K. Kanaris, and K. Kotis, “Automatic generation of multiple choice questions from domain ontologies,” *MCCSIS’08 - IADIS Multi Conf. Comput. Sci. Inf. Syst. Proc. e-Learning 2008*, vol. 1, no. January, pp. 427–434, 2008.

[15] Y. S. Chen, D. M. Ran, and X. H. Zhu, “Automatic Generation of Standard Examination Questions Based on Ontology,” *Appl. Mech. Mater.*, vol. 373–375, pp. 1747–1750, 2013, doi: 10.4028/www.scientific.net/amm.373-375.1747.

[16] M. Divate and A. Salgaonkar, “Automatic question generation approaches and evaluation techniques,” *Curr. Sci.*, vol. 113, no. 9, pp. 1683–1691, 2017, doi: 10.18520/cs/v113/i09/1683-1691.

[17] A. Thomas, P. Frank-Bolton, T. Stopera, and R. Simha, “Stochastic tree-based generation of program-tracing practice questions,” *SIGCSE 2019 - Proc. 50th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 91–97, 2019, doi: 10.1145/3287324.3287492.

[18] Z. Fan, Z. Wei, S. Wang, Y. Liu, and X. Huang, “A Reinforcement Learning Framework for Natural Question Generation using Bi-discriminators,” *Coling*, pp. 1763–1774, 2018, [Online]. Available: https://aclanthology.info/papers/C18-1150/c18-1150.

[19] T. H. Tan, “PolyAQG framework: auto generating assessment questions,” *[Unpublished Dr. Diss.*, Sunway Uni, 2022.