International Journal of Artificial Intelligence in Education (2024) 34:332–375
https://doi.org/10.1007/s40593-023-00333-6

ARTICLE

A Novel Framework for the Generation of Multiple Choice Question Stems Using Semantic and Machine-Learning Techniques

Archana Praveen Kumar[1] · **Ashalatha Nayak**[1] · **Manjula Shenoy K**[2] · **Chaitanya**[1] · **Kaustav Ghosh**[1]

Accepted: 27 February 2023 / Published online: 30 March 2023

Abstract
Multiple Choice Questions (MCQs) are a popular assessment method because they enable automated evaluation, flexible administration and use with huge groups. Despite these benefits, the manual construction of MCQs is challenging, time-consuming and error-prone. This is because each MCQ is comprised of a question called the "stem", a correct option called the "key" along with alternative options called "distractors" whose construction demands expertise from the MCQ developers. In addition, there are different kinds of MCQs such as Wh-type, Fill-in-the-blank, Odd one out, and many more needed to assess understanding at different cognitive levels. Automatic Question Generation (AQG) for developing heterogeneous MCQ stems has generally followed two approaches: semantics-based and machine-learning-based. Questions generated via AQG techniques can be utilized only if they are grammatically correct. Semantics-based techniques have been able to generate a range of different types of grammatically correct MCQs but require the semantics to be specified. In contrast, most machine-learning approaches have been primarily able to generate only grammatically correct Fill-in-the-blank/Cloze by reusing the original text. This paper describes a technique for combining semantic-based and machine-learning-based techniques to generate grammatically correct MCQ stems of various types for a technical domain. Expert evaluation of the resultant MCQ stems demonstrated that they were promising in terms of their usefulness and grammatical correctness.

Keywords Ontology · Multiple choice questions · Machine learning · Semantic web · Semantic web rule language · Description logic

✉ Ashalatha Nayak
asha.nayak@manipal.edu

Extended author information available on the last page of the article

Introduction

The use of Multiple Choice Questions (MCQs) as an assessment tool is gaining more attention in the current education field (Yaneva et al., 2018) Some of the benefits of using MCQs are that they: a) offer the opportunity to measure intelligence, knowledge, or cognitive skills, b) are easy to evaluate, and c) can be administered to huge groups. Due to these advantages, MCQs have been popularly used to aid decision-making during job placements, and college admissions. In addition, MCQ stems can also evaluate whether the relevant course outcomes of a particular course have been satisfied, which can help to review and revise the instructional activities if needed.

Despite these benefits, there are challenges associated with developing and using MCQs. One of these is the need to develop many distinct MCQ stems for each course (Haladyna & Rodriguez, 2013; D'Sa & Visbal-Dionaldo, 2017). According to Wood (2009), the answers can be memorized by reusing the same MCQ stem, thus posing a threat to the validity of the exam. Furthermore, for a given course, the design of MCQ stems should be able to address the course objectives along with course outcomes (Tarrant & Ware, 2012). Hence, the manual construction of MCQ stem is time-consuming, cumbersome, and error-prone (Hansen & Dexter, 1997; Tarrant et al., 2006). Most MCQ stem developers have inadequate expertise and training to develop high-quality MCQ stem (Tarrant et al., 2006). These item writing flaws lead to the construction of ambiguous MCQ stem, which does not ensure the validity of the questions as established by the expert panel review (Considine et al., 2005; Xie et al., 2022). According to Considine et al. (2005), the validity testing of MCQ stem concerning content can be established only by the expert panel review.

To solve all the problems mentioned above, technology in automatically generating MCQ stem holds much promise. Rus et al. (2008) defines Automatic Question Generation (AQG) as a task to develop the questions automatically from various inputs such as text, database, or semantic representation. Hence, automatically generated questions assist in measuring the learning capability and offer a quicker solution to large-scale assessment tests (Gierl et al., 2017). The construction of MCQ stems through AQG also facilitates the usage of MCQs in drills and practice sessions without much problem. AQG can also be customized to design personalized MCQs for test takers, which include their preferences and learning ability (Mostow & Chen, 2009; Shah et al., 2017).

According to Le et al.(2014), AQG, in the most recent research, deals with techniques to generate questions from knowledge resources that are either structured (e.g., ontology) or unstructured (textual data). The approach using structured knowledge resources is called a semantic-based or ontology-based approach. In contrast, Machine-Learning (ML) based or text-based approach uses unstructured data resources. Ontology generates MCQ stem due to its semantics and precise syntax to represent the domain knowledge (Alsubait, 2015). This knowledge representation then describes the question stem. Therefore semantic-based approaches have generated heterogeneous MCQs in various domains using structured knowledge resources. ML-based techniques have also gained popularity where classifiers trained with textual data features identify the relevant sentences (Kurdi et al., 2020) that are converted into primarily only Cloze questions.

In the existing literature, it is observed that AQG towards MCQs has been generated predominantly for the language learning domain (Alsubait, 2015). The contribution of AQG has been minimal towards the technical domain (Heilman, 2011). In the current education system, the technical domain use MCQs extensively (O'Dwyer, 2012). According to Narayanan et al. (2015), MCQs on engineering education need to comprise questions that: a). are real-life problem solving and inductive learning with reasoning, b). can satisfy the cognitive skills based on Bloom's Taxonomy, and c). can satisfy the required course outcomes for a given course. Constructing MCQs manually that fulfill these guidelines requires tremendous effort (Testa et al., 2018). Therefore this research is an attempt to generate MCQ stems automatically for a technical domain based on Bloom's taxonomy cognitive levels (Testa et al., 2018) to structure and characterize the assessment in terms of complexity and higher order skills.

AQG in the current research has generated a variety of MCQs like Cloze and Wh-type questions (questions starting with 'Where', 'What', and many more.). However, based on observations by Kurdi et al. (2020), ontologies fail to generate grammatically correct Cloze questions compared to ML techniques. The main reason is that verbalizing ontology into Cloze questions generates grammatically incorrect questions (Faizan & Lohmann, 2018). In addition, the AQG method need not infer any semantic reasoning towards developing Cloze questions. Hence the unstructured data can be suitably used to generate Cloze questions using ML. Based on the review by Ch and Saha (2018), ML generates reasonably good Cloze questions but fails to generate grammatically correct Wh-type questions compared to the ontology technique. Given the above issues and limitations, it is imperative to develop a system that can generate MCQ stems automatically for a technical domain. Hence the objectives are formalized into the following research questions:

1. RQ1: Can a system automatically generate different Wh-type and Cloze question stems for a technical domain?
2. RQ2: Can this system generate MCQ stems that assess cognitive skills as categorized in Bloom's Taxonomy?
3. RQ3: Can this system generate useful and grammatically correct Wh-type and Cloze MCQ stems using a hybrid combination of ontology and ML?

This research proposes a hybrid approach using an Ontology-Based Technique (OBT) and Machine-Learning Based Technique (MBT) to generate different types of Wh-type and Cloze question stems for a technical domain. The research proposes the following objectives:

– A hybrid framework of OBT and MBT to generate heterogeneous MCQ stems for a technical domain

– Generates Wh-type question stems using OBT and Cloze question stems using MBT
– Evaluates MCQ stems based on Bloom's Taxonomy

The rest of the article is subdivided as follows: Background is presented in "Background" section. "Related Work" section explains the Related Work. The methodology is shown in "Methodology" section, and "Results of Experiment and Analysis" section provides the experiment and analysis results. Evaluation of the system is discussed in "Evaluation" section, and the conclusion is detailed in "Conclusion" section.

Background

Predominantly semantic-based approaches utilize ontology and its components for the automatic generation of MCQ. This section briefly introduces ontology, MCQ, different types of MCQ, and the cognitive skills classified based on Bloom's Taxonomy.

Ontology

Gruber (1993) defines ontology as a concrete specification of the domain. Ontologies represent knowledge that can be used as a foundation to build many intelligent applications. Recent advances in publishing knowledge in the form of ontologies have led to the increased use of these structures in educational applications (Vinu & Kumar, 2015). Researchers use existing or hand-crafted ontologies for a given domain to generate assessment questions. Ontology can be either built by (a). Using open source software Protégé Stanford Center for Biomedical Research (2019) or (b). Using a programming language, i.e., Web Ontology Language (OWL). Ontology provides an explicit specification of a domain modelled through the ontology components of concepts, instances, attributes (datatype property), relations (object-type property), and axioms Gruber, 1995).

Concepts are classes; instances are individuals of a concept; relations are attributes of a concept or relationships between concepts; axioms are restrictions or constraints on the concepts. Description Logic (DL) from the family of logic-based knowledge representations conveys the assertions or facts on concepts, instances, and relations through axioms of the ontology (Horrocks, 2005). The axioms added are called Terminological axioms (TBox) and Assertional axioms (ABox). According to Grosof et al. (2003), TBox is used to structure the domain, i.e., the schema of the domain, while ABox shows the instances of the domain. Hence ontology is analogous to database models except that the database reflects data in tables, while ontology reflects data in a knowledge graph. Additional higher-level constraints or rules to specific roles satisfied by concepts extend the ontology's semantics (Eiter et al., 2008). According to Horrocks et al (2004), Semantic Web Rule Language (SWRL) asserts facts such as: 'Father is a male having a human child' or 'Parent': as an inverse relationship of a ’child’. SWRL is a rule-based markup language comprising a set of rules with antecedent and consequent. The rule implies that whenever the conditions in the antecedent hold, the conditions in the consequent must also hold.

Consider an example: For a university domain, the three classes or concepts are—Person, Faculty, and Student. A Person can be either a Faculty or a Student. So Faculty and Student are sub-classes under Person. Each Faculty has a datatype property—Name, Age, Address, EmpId, Dept, and Salary. Each instance of Student will have datatype property—RegNo, RollNo, Section, Name, and DOB. Faculty teaches Students, so ’teaches’ is an object-type property. X is an instance under Faculty, while Y is under Student. Figure 1 shows the ontology for this domain done through Protégé.

DL also provides reasoning capability along with constructs of conjunction, disjunction, negation, and quantifiers to the ontology (Baader et al., 2005). Hence ontology is said to represent the domain at the semantic level. A built ontology can infer certain assertions and add them through reasoning techniques. Thus ontology not only represents knowledge but also adds and extends semantics by inferring through reasoning. Due to this semantic representation, research uses ontology models to generate MCQs automatically.

MCQs

An MCQ introduces the question called stem (Majumder & Saha, 2015) and has one correct answer called the key, with three to four wrong options called distractors. It is either a single-response question with only one key or a multiple-response question having multiple keys. The approach in this research is towards the automatic generation of a single-response MCQ. Figure 2 shows the structure of single response MCQ. The MCQ questions can be in different variations like Wh-type,

Fig. 1 Sample of university ontology

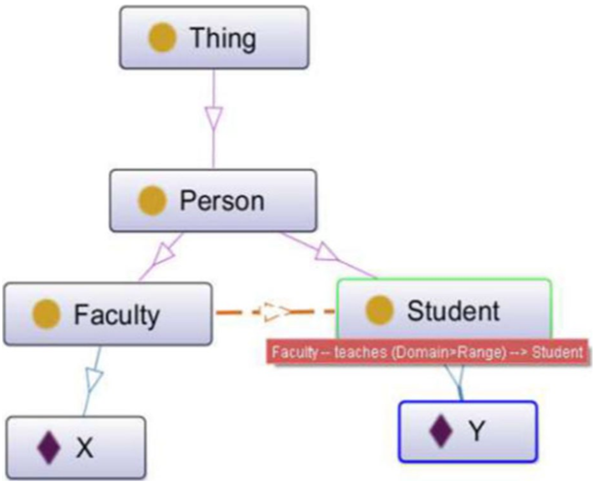
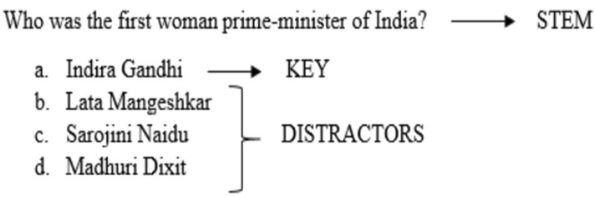


Fig. 2 Sample of MCQ



Definitions, One-word answers, Synonyms, T/F (True/False), Odd one-out, Analogy, and Cloze questions (Agarwal, 2012).

Interrogative Questions which start with Wh-word, e.g., 'When', 'Who', 'Where', 'Why', 'Which' and 'What' are Wh-type questions. 'What' is mostly used in questions about a term directly or indirectly. Direct questions of the form 'What is X?' represents Definition questions. Questions referring to the concept indirectly of the form 'What is the concept which yields X?' answer one word. Henceforth, such questions are referred to as One-word answer questions. Questions to extract the equivalent name of the given concept are synonyms, e.g., 'What is the equivalent name of X?'. The True/False questions determine whether a given stem is true or false. Odd one-out questions are Wh-type questions to identify the key which satisfies the stem, e.g., 'Which among the following is of type X'?. Analogy questions involve giving a special relationship and then identifying another similar one. E.g.: 'As Fuel: Car then Food: ?'. Cloze questions are questions where a word is substituted by a blank, e.g., 'A part of the word from the sentence is termed as _________'. A given MCQ stem needs to satisfy a certain cognitive skill based on Bloom's Taxonomy (Dunham et al., 2015).

Bloom's Taxonomy

Bloom (1956) identified the three learning domains or educational activities: Cognitive Knowledge or Mental Skills, Affective Attitude or Emotions, and Psychomotor or Physical Skills. Technical domain education requires questions assessing intellectual skills such as problem-solving and critical thinking belonging to the cognitive domain (Anderson & Krathwohl, 2001). This research aims to generate such question stems automatically. So this section introduces only those educational objectives under the cognitive domain.

Under cognitive, the six skills are remembering, understanding, applying, analyzing, evaluating, and creating (Palmer & Devitt, 2007). Table 1 shows the different levels of cognitive learning (Krathwohl, 2002). For the learning domain, the MCQ

Table 1 Cognitive skills as stated in Bloom's taxonomy (Krathwohl, 2002)

Taxonomy levels	Cognitive skills
Level I	Remembering –Recall information
Level II	Understanding and Applying – Comprehending and being able to interpret the data
Level III	Analyzing, evaluating and creating – Recalling and using the information to solve the new problem

needs to evaluate all the student's possible skills for the given course (Narayanan et al., 2015). However, MCQs are not appropriate for testing higher levels of creativity (Carneson et al., 1996). Nevertheless, the MCQ can test the other higher levels of evaluating, applying, and analyzing, but often it tests the lower order learning of understanding and remembering (Carneson et al., 1996). There are certain question words used in MCQ to test the different cognitive skills (Anderson & Krathwohl, 2001) shown in Table 2.

Related Work

AQG has dominated for three decades using either unstructured or structured knowledge resources (Effenberger, 2015). According to Alsubait (2015), the usage of either ontology-based or ML techniques has been prevalent and predominant in the automatic generation of MCQ. Most ML techniques have targeted generating Cloze questions in the language learning domain. The ML approaches quickly generate Cloze questions due to the high similarity between the generated questions and the input text, (Rakangor & Ghodasara, 2015). Furthermore, the distractors of Cloze questions are words with different spelling or grammatical sense of key (Brown et al., 2005). Leo et al. (2019) agrees and mentions that MCQs other than Cloze questions require semantics with varying syntactic structures of the text. In this context, ontology-based approaches have been successful and relevant towards generating meaningful Wh-type questions (Ch & Saha, 2018). Hence, the related work discusses ontology and ML approaches for generating MCQs.

Most ML approaches focus on generating Cloze questions. To generate Cloze questions first step requires informative sentences to be identified. The next step is identifying a target word or key in the informative sentence. The last step is modifying the informative sentence by replacing the blank with the target word generating the Cloze question. The first step needs domain-based features for selecting informative sentences. Most approaches use conditional probability to identify the key in the second step. Therefore, the discussion is limited to only the first step for each related work in the ML approach.

Pino et al. (2008) presented a strategy to select sentences that improvised the quality of the Cloze questions. Here the input sentences having the target word were

Table 2 Question words in MCQs to infer the cognitive skills (Anderson & Krathwohl, 2001)

Skills	Question words used
Creating	Integrate, design, modify, construct, create
Evaluating	Appraise, judge, evaluate, assess, recommend
Analyzing	Analyze, arrange, order, compare, infer
Applying	Apply, compute, demonstrate, show, relate, use
Understanding	Interpret, discuss, predict, summarize, classify
Remembering	List, define, label, describe, name, match, cite, reproduce

chosen as informative sentences. Each sentence was given weights based on features: grammar, context, complexity, and length of sentence. Relevant sentences were those with scores greater than the threshold and converted to Cloze questions for English vocabulary assessment.

Correia et al. (2012) made AQG efforts to generate Cloze questions in Portuguese. In contrast, Aldabe et al. (2009) attempted similar research in the Basque language. Both researches built a gold standard corpus with the aid of domain experts in the respective languages and trained Support Vector Machine (SVM) on the corpus features to identify the informative sentences. Features chosen comprised: sentence length (Pino et al., 2008), the position of the target word, proper nouns, foreign words, co-occurrences, verbs, acronyms, and numerical expressions. Once trained on this set of features, SVM filtered and identified potential informative sentences from the input sentences. These researches tested which features could be used for SVM training to output the best informative sentences for a given input corpus for a language other than English.

Cloze questions from biology textbook were generated by Agarwal and Mannem (2011) wherein informative sentences were extracted based on specific features. Words were common in the chapter title and sentences of that chapter in the biology domain. So the features used were the count of nouns and adjectives in the sentences, identical words in the title and sentence, abbreviations, and the position of the word in the sentence. Based on the feature satisfied, each sentence was assigned a weight. Relevant sentences were those with scores greater than the threshold and converted to Cloze questions. The limitation was that some features in the approach generated irrelevant Cloze questions.

Effenberger (2015) parsed input sentences and proposed different features to extract the informative sentences of news articles to transform them into Wh-type questions. The features comprised: the number of occurrences of the target word, the sentence contained in the headline, and the target word's depth in the parse tree. Depending on the feature satisfied, each sentence was assigned a weight. The relevant sentence was the one with a score greater than the threshold. The approach generated only Wh-type questions from the relevant sentences that were either too easy or ambiguous. Majumder and Saha (2015) proposed a technique for selecting informative and relevant sentences from Wikipedia using topic modeling for the sports domain. This approach extracted the sentences having the required topics. Then structural similarity between the syntactic parse tree of the input sentences was compared with the syntactic parse tree of the reference sentences. Reference sentences were those compiled and extracted from MCQs of the sports domain. Relevant sentences were input sentences with similar syntactic parse trees to the reference sentences. The sports domain comprised words about different locations, sports person names, tournament names, and trophies won. These domain-related words were the target words, and the Wh-type question was generated based on the identified key. The limitation of the approach was the generation of irrelevant Wh-type questions due to the false detection of topics from Wikipedia.

Using the Fireflies Algorithm and Preference Learning, Sahathanavijayan et al. (2017) proposed an approach to generate Cloze questions from web pages. In this approach, a user-defined query retrieved all the web pages searched by Google search API. Then only the text within the HTML para tag was identified and extracted. This textual data was summarized and optimized using preference learning and the fireflies algorithm. The summarized information comprised only those relevant sentences based on specific features of 1. the sentence length (Pino et al., 2008), 2. sentence without pronouns and adverbs, and 3. frequent co-occurring words. After tokenizing all words in these relevant sentences, only cardinal words or pronouns or describing adjectives were substituted as blanks to generate Cloze questions. The limitation of the approach was in cases where summarized sentences containing pronouns were not resolved to their corresponding nouns, that generated irrelevant questions.

In the semantic-based approach, the researchers used ontology and its components, namely DL, SWRL, or SPARQL, to generate the different questions. Researchers used existing or built ontology to generate the required assessment tool. Initially, Papasalouros et al. (2008) made use of an ontology to generate MCQs by suggesting eleven strategies based on class, property, and terminology. The research needed more technical details to clarify which strategy had to be used to generate the different types of MCQ. Furthermore, the generated MCQ comprised the questions with the stem -'Choose the correct sentence'. Stasaski and Hearst (2017) generated MCQs by randomly choosing a concept and exploring its outgoing links in an ontology. The concept and its outgoing links generated Wh-type questions with the individuals having specific data or object-type properties. This research generated factual questions with different MCQ stems but experimented only with the properties of the ontology.

In the medical domain, Leo et al. (2019) used an ontology to generate a case-based multi-term MCQ whose answer was the diagnosis of the disease for a particular patient. The research used The Elsevier Merged Medical Taxonomy (EMMeT-OWL) ontology (Parsia et al., 2015), which had all the details about the terms, relations of clinical concepts, and annotations. The generated questions were either too easy or too difficult, rated by the reviewers, and therefore could not be used in the medical exam.

Some researchers like Cubric and Tosic (2011) had extended domain ontologies by adding annotations and semantic interpretations between the target question ontology and domain ontology. The prototype discussed in this research was an extension of Holohan et al. (2005) work, which generated Wh-type stems of knowledge, application, and analysis level under Bloom's Taxonomy. The annotations and semantic interpretations were restrictions on the ontology, which helped to generate MCQ. One more researcher Jelenkovi and TOˇSI (2015) implemented an automatic MCQ generator using an ontology called OpenSeMCQ. Their implementation generated Wh-type questions to test the domain knowledge but needed high-quality questions, and neither were suitable for actual use. Alsubait (2015) and Alsubait et al. (2012) utilized TBox axioms of an ontology to generate MCQs in the direction of AQG. The former research generated definition questions, while the latter generated analogy questions. Both approaches generated questions to assess the factual cognitive level only.

Venugopal et al. (2016) proposed a method by exploiting SPARQLrules and templates to frame questions of the type 'Choose an X which has object-type property with

Y and data property Z?'. Venugopal and Kumar (2015) exploited ontology's DL queries in terms of TBox and ABox axioms generating Wh-type questions and Cloze questions for a non-technical domain. Along with these works, some researchers like Zoumpatianos et al. (2011) made use of SWRL to create the Cloze questions. The above three approaches defined the semantics of the question but needed to generate syntactically correct questions.

Summary of Related Work

Tables 3 and 4 show the existing works in ontology and text-based approaches discussed above. From Tables 3 and 4 the following inferences can be derived:

– Majority of the approaches have concentrated only on non-technical domains and therefore have no relevance in evaluating questions based on Bloom's Taxonomy
– Among the approaches that focussed on the technical domain have generated Wh-type questions limited only to cognitive Level I
– Furthermore, the existing ontology approaches which have generated Cloze questions are grammatically incorrect compared to the Cloze questions generated from the ML approach

To summarize, in the existing literature, there were many efforts towards the automatic generation of MCQ; but only a few attempts were made towards developing MCQ in the technical domain that satisfied Bloom's Taxonomy. The text-based approaches generated grammatically correct Cloze questions, but most were made towards the language learning domain. These approaches used essential features to identify the relevant sentences based on the grammar constructs of the language. Moreover, the transformation of relevant sentences to Cloze questions required only the substitution of a blank in the original sentence, so the questions were grammatically correct. Furthermore, Cloze questions, if generated by ML for the technical domain, would not require evaluation for grammatical correctness, thereby reducing the cost of evaluation (Das & Majumder, 2017; Mostow & Jang, 2012). However, it is challenging to identify the features required to extract the relevant sentences for a technical domain from textual data. The semantic-based approach generates semantically rich questions from different ontology constructs of DL or SWRL. However, research to utilize both DL and SWRL constructs of ontology to generate different MCQs for a technical domain has yet to be attempted. This research is a unique attempt to generate the different MCQs using a structured representation of ontology and unstructured textual data to generate Cloze questions for a technical domain. In addition, the research has also been evaluated for satisfying the cognitive skills based on Bloom's Taxonomy. Furthermore, the domain experts have evaluated the generated questions based on the required metrics.

Table 3 Existing ontology based approaches

Reference	Input	Domain	Question type	Bloom's taxonomy	Strategy	Remarks
(Papasalouros et al., 2008)	Ontology	Not specific	MCQ	Not done	Used classes, properties and terminology of ontology to generate questions	The MCQ stem generated only of the type 'Choose the correct sentence'
(Zoumpatianos et al., 2011)	Ontology	Not specific	Cloze	Not done	Used SWRL to generate questions from ontology	Cloze questions were not grammatically correct
(Venugopal & Kumar, 2015)	Existing Ontology	Not specific	MCQ	Not done	Proposed two new techniques to generate label sets of instances with Open World Assumptions. These label sets were used to generate MCQ	There were some questions which were not grammatically correct
(Alsubait, 2015)	Hand crafted Ontology on Java and Knowledge Acquisition	Technical	Wh-type	Recall Level	Computes similarity between the concepts of ontology and based on the values generates questions of various difficulty level	All generated questions satisfy cognitive Level I
(Leo et al., 2019)	Existing EMMET ontology	Medical	Case based MCQ	Not done	Uses classes, relations and annotations of the ontology. Questions were framed using templates	Questions generated not suitable for real time applications

Table 4 Existing machine-learning approaches

Reference	Input	Domain	Question type	Bloom's taxonomy	Strategy	Remarks
(Pino et al., 2008)	Text	English Vocabulary assessment	Cloze	Not done	Used dictionary and added linguistic features to select relevant sentences	The dictionary has one sentence for the context compared to the corpus thereby certain relevant sentences would be discarded
(Agarwal & Mannem, 2011)	Text	Biology	Cloze	Not done	Used features of count of nouns, adjectives, and words common in the abbreviations to select relevant sentences	Some generated Cloze questions were irrelevant, so needs fine tuning of certain features
(Majumder & Saha, 2015)	Text	Sports	MCQ	Not done	Used topic modelling to extract sentences from Wikipedia and informative sentences were identified based on similarity of input syntactic structure with parse tree of existing MCQ stem	Topics were identified by an open tool so certain topics detected were false which extracted irrelevant sentences
(Sahathanavijayan et al., 2017)	Text	User defined	Cloze and Analogy	Not done	Relevant sentences were extracted from web pages based on user query. From these pages sentences were selected based on completeness and importance	Sentences selected based on completeness had not included sentences having pronouns thereby certain relevant sentences with pronouns would be missed

Methodology

The proposed system utilizes both structured and unstructured knowledge resources to solve the research objectives. In this context, the following section explains the proposed system's overview, a working example, and the algorithms implemented for solving the problem.

Overview of the Proposed System

Figure 3 shows the proposed system to generate a single response MCQ stem automatically. The input PDF file is preprocessed automatically through a program to obtain the text file. The steps of the program, which reads the input PDF file and converts it to a text file includes the following:

1. Convert PDF file into DOCX (Microsoft Word Open XML Format Document) and read the DOCX file
2. Identify the different sections of the file through XML tags
3. Remove the irrelevant sections (figures, tables, exercises) and sentences referring to the irrelevant sections
4. Segregate paragraphs into individual sentences using Natural Language Toolkit (NLTK)
5. Convert the DOCX file into a text file
6. Send the text file to subsequent stages of OBT and MBT simultaneously

NLTK comprises a library of functions that perform Natural Language Processing (NLP) tasks on the input sentences, such as tokenization, chunking, and sentence segregation. The following sections give details on OBT and MBT to generate the different MCQs.

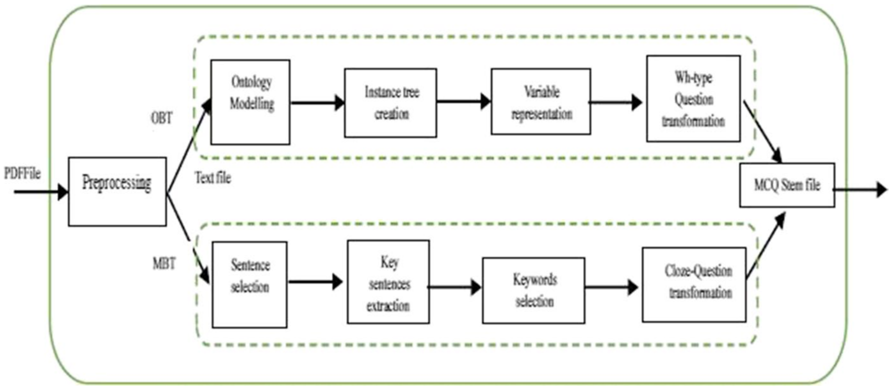


Fig. 3 Proposed system

Ontology Based Technique – OBT

OBT, which generates Wh-type questions, involves three stages: Ontology modeling, Instance Tree (ITree) creation, with the last stage of Variable Representation and Wh-type Questions transformation.

Ontology Modeling Encompasses steps to build an ontology with its components: concepts, relations, instances, and axioms. With no existing ontology available on the given domain an ontology is built using Protégé software semi-automatically. For this construction, each sentence of the preprocessed text file is read and converted into a triple format of < Subject, Predicate, Object >. This conversion is done through the NLTK toolkit automatically for every segmented sentence. The subject and object of each sentence form the concepts of ontology. The predicates represent the properties of the ontology. DL or SWRL rules add additional facts, instances, equivalent classes, and subset classes of the sentences into the ontology.

Every rule comprises of antecedent and consequent. A rule represents concepts or instances along with properties in the Left Hand Side (LHS), also called antecedent, or Right Hand Side (RHS), called consequent. This research uses DL rules to add TBox and ABox axioms. DL rules represent either a subset operator (⊆ symbol indicating a member of a set) or an equivalent operator (≡ symbol denoting equivalent classes) to depict the required TBox axioms. Equivalent classes represent two classes with different names but similar attributes. ABox axioms add the instances, datatype, and object properties into the ontology. Once added, constraints to be satisfied among classes are expressed using SWRL rules. SWRL rules use the implication operator (→) to represent the constraints satisfied by the antecedent giving the consequent. The naming convention of classes and properties follows a pattern as required for the question generation explained later.

SWRL or DL rule represented as SR in Eq. 2 comprises a set of variables VR_p where p ranges from 1 to j. Each $VR_p = (At_p, Ct_p)$ where At_p is antecedent and Ct_p is consequent, which consists of classes or properties as shown in Eq. 1. The At_p and Ct_p are joined by either implies (→) in SWRL or equivalence (≡) or subset (⊆) in DL. Equation 2 shows how each rule SR is a combination of VR_p used along with the operators of → or ⊆ or ≡.

$$VR_p = \left(At_p, Ct_p\right) where\ 1 < p < j \tag{1}$$

$$SR \rightarrow / \equiv / \subseteq (VR_1, VR_2, \ldots \ldots VR_p) \tag{2}$$

Here, some sample examples identified in the built ontology are explained. For simplicity and generalization, naming conventions of classes and representations of rules are denoted in a pattern. The pattern specifies the classes followed by the relations so that the pattern can be easily extended and allows the system to be input agnostic. Keeping the convention, the identified classes are 'OperatingSystem', 'Hardware', 'Interrupt', 'Software', 'SystemCalls', etc. The identified sub-classes are 'CPU' and 'Memory', both under the class 'Hardware'. 'Memory ⊆ Hardware' is a TBox axiom in the ontology. The identified TBox axiom for the equivalent class is: ’MainMemory ≡ PhysicalMemory’. Similarly, ABox axiom to add instance is: ’OperatingSystem(MS-DOS)’, for object property is: ’isTriggerredBy(Interrupt, Software)’ and for datatype property is: ’isVolatile(?Value)’ where the Value could be either true or false.

Some identified class attributes form the datatype property: ’hasSize’, ’hasSpeed’, ’isVolatile’ for class ’Memory’. Relations or object-type properties represent the inter-class relationships between classes and individuals. E.g. ’controlsCoordinates is an inter-class relation between classes OperatingSystem and Hardware’. To represent a fact: ’Interrupt triggered by software is a system call’, SWRL constraint is added. This constraint is given by ’Interrupt(?i), Software(?s), isTriggeredBy(?i, ?s) → SystemCalls(?i)’ in SWRL. The rule comprises classes ’Interrupt’, ’Software’, and the object type property ’isTriggeredBy; on the antecedent side. While the consequent side comprise of another class ’SystemCall’. The alphabets in SWRL rule symbolize the instances of the ontology classes. Once all the ontology components along with axioms are added, the rules are passed to the second stage of Instance Tree Creation in OBT.

Instance Tree (ITree) Creation Each DL and SWRL rule gets transformed into its corresponding ITree. Hence for ’n’ rules, this step yields ’n’ ITrees. According to the consequent, this ITree is converted into a Wh-type question. Initially, a root node for the consequent is created in this step. The root node is named with the instance of the consequent. Then next, the child node having the classes and relations for the same instance is added under the antecedent and consequent. The following section shows how ITree is generated for the sample SWRL and DL rules.

Sample SWRL Rules and Corresponding ITree Creation.

SWRL Example: Hardware(?j) ∧ hasControlOver(?j, ?i) ∧ InputOutputDevices(?i) → DeviceController(j)

The above rule adds a fact for the sentence ’THE HARDWARE THAT HAS CONTROL OVER INPUT OUTPUT DEVICES IS CALLED DEVICE CONTROLLER’.

In the rule given above, the class variables on the antecedent side are Hardware and InputOutputDevices, while DeviceController is a class variable on the consequent side. Here i and j are instances while ’hasControlOver’ is an object-type property/inter-class relationship variable between the instances i and j. Initially, the instance under the consequent is taken and its corresponding antecedent and consequent are created as a root node. So root node with instance ’j’ is created. Under the antecedent of ’j’, the rule comprises class variable ’Hardware’ and property variable ’hasControlOver’ with instance ’i’. While, under the consequent of instance ’j’, the rule comprises class variable ’DeviceController’. As ’hasControlOver’ has a relationship with ’i’, ITree for ’i’ is created and added through steps similar to ITree of ’j’. The diagram of ITree of **Example 1** is created as shown in Fig. 4.

SWRL Example 2: AccessMethods(?m) ∧ hasSequentialAccess(?m, true) → SequentialAccess(?m)

The rule states the fact for the sentence ’ACCESS METHOD WHICH HAS SEQUENTIAL ACCESS IS CALLED SEQUENTIAL ACCESS’. The rule provides the fact for only one instance ’m’ for which the ITree is created. The ITree creation

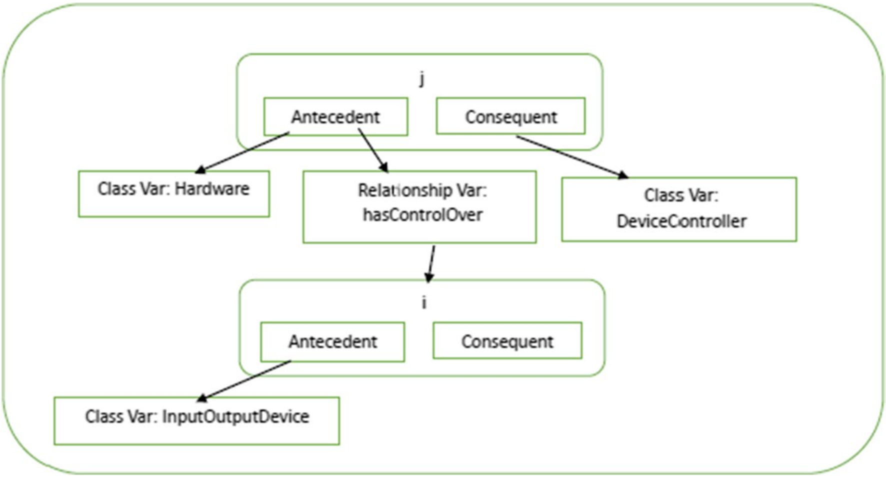


Fig. 4 Sample instance tree for SWRL Example 1

for this SWRL rule follows the same steps mentioned for SWRL Example 1. Figure 5 shows the ITree generated for **Example 2.**

SWRL Example 3: SecondaryMemory(?m) → Volatile(?m, false)

This rule states the fact in the sentence 'SECONDARY MEMORY IS THE MEMORY WHICH IS NOT VOLATILE'. This rule is an example of using data type property on the consequent side of the SWRL rule. Figure 6 shows the ITree generated for Example 3.

Sample DL Rules and corresponding ITree creation

DL rules represent fact for either equivalent or subset classes. So here, the steps are followed depending on the operation used. In the case of equivalence, as all instances satisfy the constraints of the classes, so DL rule does not explicitly specify the instances in the expression. Therefore in the ITree for DL, 'e' is chosen as an arbitrary class instance as the root node. On similar lines, 'a' is chosen as an arbitrary class instance as a child node. Equivalence is shown by representing the

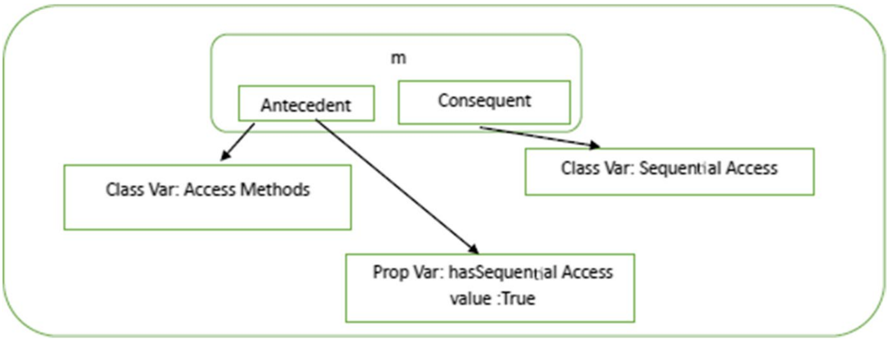


Fig. 5 Sample instance tree for SWRL Example 2

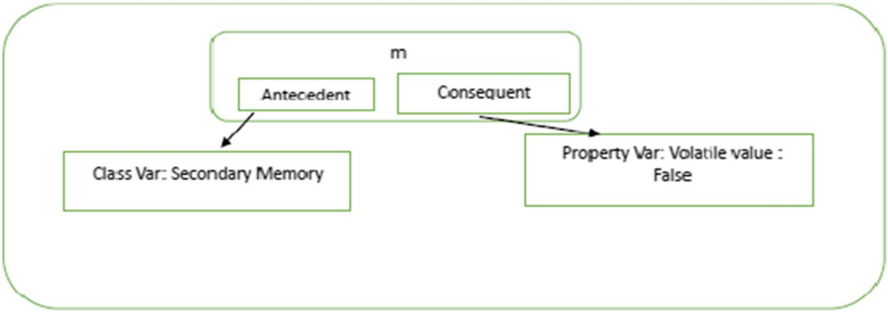


Fig. 6 Sample instance tree for SWRL Example 3

concept equivalent to the class variable through the relationship variable 'theEquivalentNameOf'. In the case of representing a subset, the concept of the consequent is mentioned as a type variable. Type variable is a type set comprising of different types under the antecedent of the root node 'e'.

DL Example 1: ***ShortTermScheduler*** $\equiv$ ***CPUScheduler***

The above example is a DL rule that states a fact for the sentence 'SHORT TERM SCHEDULER IS ALSO KNOWN AS CPU SCHEDULER'. In the rule given above, there are class variables on both the antecedent and the consequent sides. This rule is an example of equivalence, so here the instance tree is created with 'e' and 'a' as mentioned above. The diagram of ITree creation for Example 1 is shown in Fig. 7.

DL Example 2: BatchOperatingSystem $\cup$ RealTimeOperatingSystem $\cup$ NetworkOperatingSystem $\subseteq$ TypesOfOS

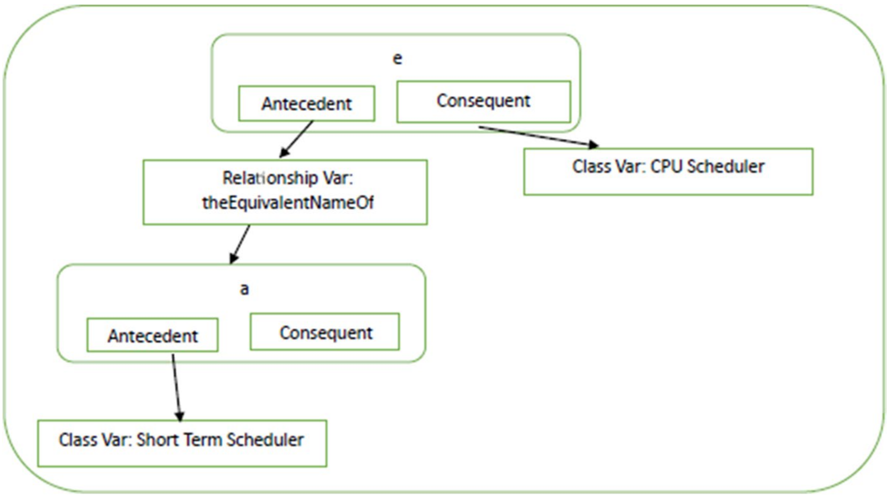


Fig. 7 Sample instance tree for DL Example 1

The above example states fact for the sentence 'BATCH OPERATING SYSTEM, REAL TIME OPERATING SYSTEM, NETWORK OPERATING SYSTEM ARE DIFFERENT TYPES OF OS'. The DL states the fact using $\cup$, i.e., the union operator. So here, the ITree has a type set denoted in the antecedent part of the root node. This rule is an example of set operation, and the ITree for Example 2 is shown in Fig. 8 with e as the root node. At the end of this step, 'n' ITrees are created and passed to the third stage of Variable Representation and Wh-type Transformation in OBT.

Variable Representation and Wh-type Transformation Each created ITree is traversed in this step and based on the consequent or antecedent; different question stems are generated. The consequent of SWRL or DL rules can be either of the following:

- 'Type variable'—in case of subset operator
- 'Property variable' with 'Class variable' on the antecedent
- 'Class variable' with antecedent 'theEquivalentNameOf'
- 'Class variable' with antecedent having 'Relationship variable'
- 'Class variable' with antecedent with 'Property variable'

If the consequent is a 'Type variable', then the odd one out question stem generated is 'Which among the following is not a 'Type Variable'?'. If the consequent is 'Property variable' then the T/F question stem generated is 'What is the 'Property variable' value for 'Class variable'?. For the other rules consequent is 'Class variable' but with a different antecedent, so the question stem is generated based on the antecedent. If the antecedent is 'theEquivalentNameOf', the synonym question stem generated is 'What is the equivalent name of 'Class variable'?'. If the antecedent has a 'Relationship variable' or 'Property variable', then the child node's corresponding class variables and property variables are added to the corresponding Class_List and Prop_List. Then, question stems are generated based on the values in the Class_List

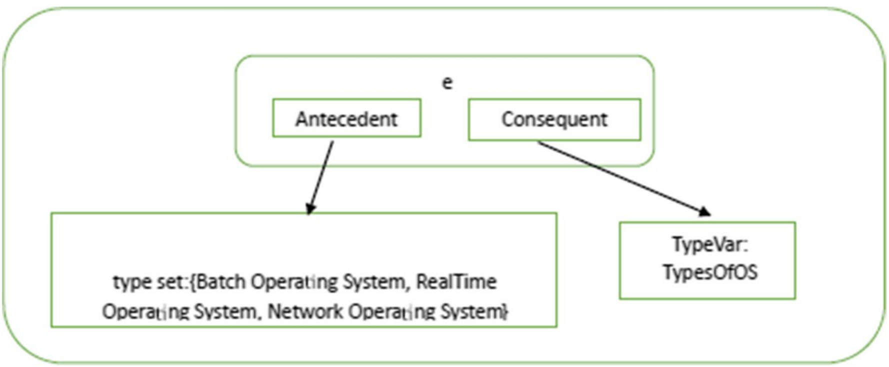


Fig. 8 Sample instance tree for DL Example 2

and Prop_List. The traversing is done from the first encountered class variable in the antecedent of the root node till the last class variable in the child node.

Class_List This is a list comprising all class variables on the antecedent side. The list is constructed progressively as a new class variable is encountered for the given SWRL rule to generate a subject clause for each question stem.

Prop_List This is a list comprising all data and object type/relationship type property variables encountered in the antecedents or consequent of the SWRL rule. This list is constructed for every new property encountered in a given rule for each class variable in the Class_List. This list helps to generate the predicate conditions for a given question stem between the two class variables.

If there is only one property variable with only one class variable in the antecedent, then the definition question stem is generated. The definition question generated is ’What is ’Class Variable’?. For cases of more than one class variable in the Class_List, a one-word question stem is generated, e.g., ’What is ’Class variable’ that has ’Property variable’ with ’Class variable’?’. The algorithm ’GenerateQuestionFromRules’ to generate the different MCQ stems from the rules is shown in the next section. This algorithm shows the second and third stages of Ontology modeling, thereby transforming ontology rules into Wh-type MCQ stems. After the algorithm, the following section shows a sample example that follows the steps mentioned in the algorithm ’GenerateQuestionFromRules’.

Algorithm

Sample Example

The sample example shows how one ITree is taken and traversed to transform the ITree into its equivalent Wh-type MCQ.**Considering the ITree Fig. 4** generated for **SWRL** Example 1, its corresponding Wh-type question is generated as follows:

Step 1: Extract the Consequent of the rule and check what is the value of consequent
Step 2: The consequent corresponds to Class Variable along with Relationship Variable with another instance
Step 3: So str_expr → ’What is ’
Step 4: Then Class_List → {Hardware}
Step 5: Appending it to the previous String Expression: str_expr → str_expr + ’a/an ’ + ’Hardware’
Step 6: Then Prop_List → {hasControlOver}
Step 7: Appending it to the previous String Expression: str_expr → str_expr + ’that ’ + ’hasControlOver’
Step 8: Again Class_List → {InputOutputDevices}

```
Input: Rules
Output: Wh-Type question stems

for each Rule R_L where 0 <  L <  n do
        Create an ITree for each instance            as shown in Section 4.2
end for
for each ITree T do
        Extract instances that refer to consequent
        if consequent = 'Type Variable' then
                Question ← 'Which among the following is not a Type Variable'?
        else if consequent = 'Class Variable' and antecedent = 'theEquivalentNameOf' then
                Question ← 'What is the equivalent name of Class Variable'?
        else if consequent = 'Property Variable' then
                Question ←'What is the Property Value for Class Variable?'
        else
                str_expr ='What is '
                create Prop_List = [ ] and Class_List = [ ]
                for each Class Variable C_U where 0 < U < q in antecedent do
                        Class_List = Class_List U C_U
                        str_expr ← str_expr + 'a/an' + C_U
                    for each Data/Relationship Property type Variable P_V where 0 < V < m in
             antecedent do
                            Prop_List = Prop_List U P_V
                            if (count(Prop_List = 1)) ∧ (count(Class_List = 1)) then
                                    str_expr ← P_V
                            else
                                    str_expr ← str_expr + ' that' + P_V
                            end if
                    end for
             end for
             Question ← str_expr + '?'
       end if
end for
```

Algorithm 1 GenerateQuestionsFromRules

Step 9: Then again appending it to the previous String Expression: str_expr → str_expr + 'a/an ' + 'InputOutputDevices'

Step 10: Finally One Word Answer question generated: WHAT IS A/AN HARDWARE THAT HAS CONTROL OVER INPUT OUTPUT DEVICES?

Similarly for the ITree in Fig. 5 corresponding to SWRL Example 4 has one Class_List and Prop_List. Therefore the lists created are: Class_List = {AccessMethod} and Prop_List = {SequentialAccess}. So a definition question is generated by the algorithm i.e. WHAT IS SEQUENTIAL ACCESS?. The algorithm generates T/F question—WHAT IS THE VOLATILE VALUE FOR SECONDARY MEMORY? for ITree in Fig. 6 corresponding to SWRL Example 5. Similarly, for the ITree in Fig. 7, the algorithm transforms the rule into a Synonym question—WHAT IS THE EQUIVALENT NAME OF SHORT TERM SCHEDULER?. Lastly, for the ITree in Fig. 8, the algorithm generates Odd one out Question -WHICH AMONG THE FOLLOWING IS NOT A/AN OPERATING SYSTEM? for DL Example 2.

Machine-learning Based Technique – MBT

The preprocessed text file after preprocessing is given to MBT to generate Cloze questions. The Cloze questions are generated in three stages: Sentence selection, Key sentences extraction, and Keywords selection.

Sentence Selection Is done by words from Topic_list, used to identify the sentences from the input text file. Topic_list is a list of relevant and important topics for a domain utilized for MBT manually created by the subject expert. The subject expert is a professor who has repeatedly taken the Operating Systems course. The subject expert identifies, picks, and keeps the topics from the manually prepared MCQ of the course in the Topic_list. This list is used in the first stage of MBT for sentence selection. The input sentence is rejected if it does not contain the topic word. This stage gives the initial list of sentences sent to the second stage.

Key Sentences Extraction The second stage of key sentence selection is done by classifying and identifying the key sentences from the sentences obtained from the first stage. For this, five classifiers are used so that the classifier which outputs the maximum number of key sentences from the total set of sentences can be chosen. As in the education domain, there is a need to generate the maximum number of relevant questions; this research has taken the maximum number of key sentences as a criterion. The classifier is trained with features used in the Feature_set. The sentences which satisfy at least one feature are chosen as Key sentence in this stage.

Feature_set Comprises of a set of features such as:

– Sentence Length (SL): Sentences of small length do not have any context, while long length would be too complex to be chosen for Cloze questions (Pino et al., 2008). The proposed research has taken the SL between 100 to 200 characters (Correia et al., 2012).
– Nouns (NN): According to Agarwal and Mannem (2011), more number of nouns in sentences increases contextual information in the sentence. Hence this feature is used to get the number of nouns in the sentence. A sentence with nouns is a good candidate for generating Cloze questions.
– Abbreviations (ABR): This is a binary feature (T/F) to check whether the sentence has ABR (Agarwal and Mannem, 2011). This feature captures whether the sentence has ABR in it. Thereby this feature determines the importance of the sentence by the presence or absence of ABR.
– Appendix Words (AW): The appendix at the end of each book comprises important words as topics or sub-topics that are used for identifying the relevant sentences. This feature is binary (T/F) to determine if the sentence has any AW from the ebook's appendix. Such sentences having AW are good candidates for generating Cloze questions.

With the features, the set of initial sentences from the MBT's first stage and the five classifiers- SVM, K-Nearest Neighbors (KNN), Decision Tree (DT), Gaussian Naive Bayes (GNB), and Logistic Regression (LR) are trained, and the classification is done. Here the classifiers are trained and tested with a 75%-25% split over preferred cross-validation. Cross-validation is preferred because it is expensive regarding the required time and computational capacity. Among the models, the most accurate classifier provides the maximum number of extracted Key_sentences. Thus the Key_sentences identified based on technical domain features for a given course are passed to the third stage of Key word selection in MBT.

Keywords Selection The third stage of keyword selection is to identify target words to be replaced by blanks, thereby transforming Key_sentences into Cloze questions. In this direction, the following steps are carried out for each key sentence: 1. Selecting a word, 2. Extracting word attributes, 3. Calculating word probability, 4. Sorting the word probabilities. Under word selection, all the words which are not stopwords are added to the Word_list. Then the attributes of each word in Word_list are extracted in the word attributes extraction. The attributes include Parts-Of-Speech (POS) of the word, whether the word is a Named Entity (NE), Syntactic dependency relation of the word in the sentence (DEP), type of NE, whether it is location, name, etc. So the attribute features of each word are added to Word_att list.

These features are converted to binary and used in the Naive Bayes classifier. The classifier accounts for the absence or presence of a feature in each word. Naive Bayes computes the probability of each word and places the word along with its probabilty into Word_prd list. Word_prd is a list comprising a word, and its probability of being replaced by blank in the Key_sentences computed based on Word_att. Lastly, the Word_prd list is sorted in descending order and is used to generate Cloze questions. Generation of Cloze question stem from each key sentence is done by substituting a blank for each word in Word_prd list. So if there are 'n' key sentences and each key sentence comprises 'm' words in the Word_prd list, there will be n * m Cloze question stems. These questions need a manual evaluation of usefulness. In this direction, the system needs to be evaluated from a correct perspective and reduce the cumbersome human evaluation. Therefore the number of Cloze questions generated from each key sentence is limited by choosing only the first three words from the Word_prd list. Thereby the system generates three Cloze questions from each key sentence. The algorithm 'GenerateClozeQuestions' to generate the different Cloze question stems from the preprocessed sentences is shown in the next section. This algorithm shows all three steps of MBT, which transforms the preprocessed sentences into Cloze question stems. After this, the following section shows a sample example that follows the steps as mentioned in the algorithm 'GenerateClozeQuestions' for one of the sample Topic_list.

Algorithm

Sample Example

Considering a sample Topic_list = {execution, binding, physical address, memory} for the given dataset, the system generates the Cloze questions as shown in the mentioned steps.

Step 1: Sentences selected

- S1—'If execution-time binding is being used, however, then a process can be swapped into a different memory space because the physical addresses are computed during execution time.'
- S2—'The relocation register contains the value of the smallest physical address'

Input: Set of Preprocessed Sentences
Output: Cloze question stems with key

for each sentence S_I where 0 < I < a **do**
    if $S_{(0)}$ ∈ Topic_list **then**
        Key_sentences ← Key_sentences U $S_{(0)}$
    end if
end for
for each Classification Model CM where M ∈ SVm, KNN, DT, LR, GNB **do**
    Train each CM with each feature in the Feature_set
    Extract the Key_sentences
    Compute the number of Key_sentences
end for
C_M with Maximum number of Key_sentences is identified as accurate classifier
Run C_M with the different dataset (smaller size than input) sentences S_J where 0 < J < a and obtain the identified Key_sentences
Start
for each Key_sentences K_D where 0 < D < p **do**
    for each W_B extract word W_B where 0 < B < m and W_B != stopword and isAlpha(W_B) **do**
        Word_list ← Word_list U W_B
    end for
    for each word W_B ∈ Word_list where 0 < B < n and n < m **do**
        Extract the attributes W_{BA}
        Word_att ← Word_att U W_{BA}
        Extract predictions W_{PRD}
        Word_prd ← Word_prd U W_{PRD}
    end for
    Sort Word_prd in descending order
    for each W_B ∈ Word_prd where 0 < B < 3 **do**
        Search the word W_B in K_O
        if Found **then**
            new_send ← K_O
            Cloze question ← new_send wherein W_B is replaced by blank
            Key ← W_B
        end if
    end for
end for
End

Algorithm 2 GenerateClozeQuestions

- S3—'When a process is allocated space, a process is loaded into memory, and a process can then compete for CPU time'
- S4—'Another possible solution to the external-fragmentation problem is to permit the logical address space of the processes to be noncontiguous, thus allowing a process to be allocated physical memory wherever such memory is available'

Step 2: Key sentences selected based on Feature_set

- S1—Selected as SL criteria met and AW present in the sentence
- S2—Discarded as SL criteria not met
- S3—Selected as SL criteria met and contains ABR
- S4—Discarded as SL criteria not met

Step 3: Key Word selection for Key sentence S1

- Word_List = {execution, time, binding, process, swapped, different, memory, space, physical, addresses, computed}

Step 4: Sorted Word_prd = {binding—0.8, swapped—0.74, addresses -0.71, process—0.69, computed -0.68, different—0.61, physical—0.6, space -0.59, time—0.58, execution—0.55, memory—0.54}

Step 5: Selecting first 3 words from Word_prd which are binding, swapped and addresses

Step 6: Cloze question generated replacing the words binding, swapped and addresses by blank

- If execution-time ________ is being used, however, then a process can be swapped into a different memory space, because the physical addresses are computed during execution time.
- If execution-time binding is being used, however, then a process can be _____________ into a different memory space, because the physical addresses are computed during execution time.
- If execution-time binding is being used, however, then a process can be swapped into a different memory space, because the physical ______ are computed during execution time.

Results of Experiment and Analysis

For the proposed approach, the course from the technical domain is used for OBT and MBT to generate Wh-type and Cloze questions combined to give the entire MCQ stems.

Dataset

The pdf file of the course is downloaded from the EBook: Operating Systems Principles (A. Silberschatch et al., 2006). The entire dataset of the book encompasses 23 chapters, each comprising roughly 30 to 40 pages, with each consisting of 20 to 30 sentences. The experiment is conducted by randomly taking four chapters comprising approximately 4500 sentences as input to the system. After preprocessing, the dataset comprising approximately 2900 sentences is input to OBT and MBT.

Experiment

OBT In this approach, using the preprocessed file, with no pre-existing ontology available online on the Operating System course, one of the authors has built the required OS ontology. The time to develop OS ontology took about 25 to 30 days. In terms of size, this OS ontology comprises 57 object properties, 10 data properties, and 344 classes. The ontology models concepts:’Computer’ and their components, such as ’ApplicationPrograms’, ’Hardware’, ’Interrupt’, ’OperatingSystem’, ’Software’ and ’Users’. Figure 9 shows the sample of the constructed ontology. As shown in Fig. 9, there are various components under the ’Kernel’ concept which depict the various terms of ’ProcessManagement’, ’MemoryManagement’, ’ProcessScheduling’, ’Protection’, ’Security’, ’TypesOfOperatingSystems’, ’SystemSoftware’, ’FileSystems’, and so on. In the ontology file, the naming convention of classes and properties follows a pattern of having capitalized initial letters of words when two different words are combined. E.g.: ’SequentialAccess’, when encountered by algorithm 1, understands the nomenclature and generates ’Sequential Access’ while generating the Wh-type question.

This section explains some of the modeled TBox and Abox axioms shown in Tables 5 and 6. Under ’Computer’ there are sub-concepts like ’ApplicationPrograms’,

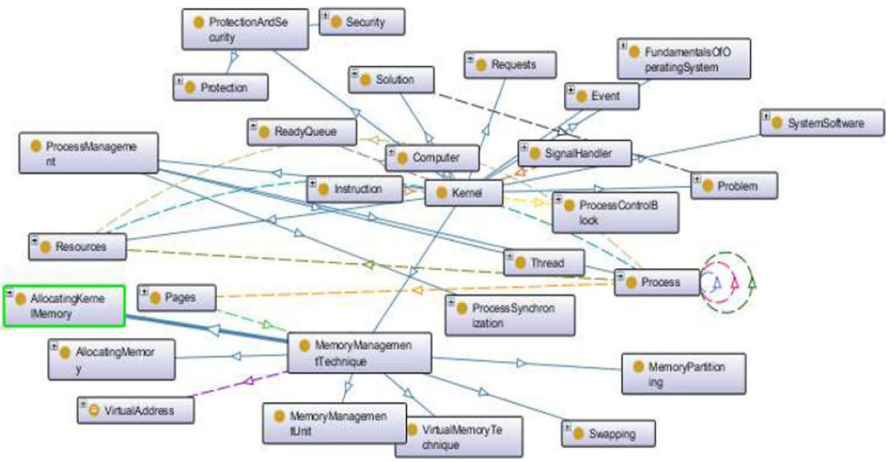


Fig. 9 Sample of operating system ontology

Table 5 Sample of TBox axioms of OS ontology

S.No	Type	Axiom
(1)	Using ⊆ TYPE A	ApplicationPrograms ⊆ Computer (a) Interrupt ⊆ Computer (b) Hardware ⊆ Computer (c) OperatingSystem ⊆ Computer (d) DirectAccess ⊆ SequentialAccess ⊆ AccessMethods (e) Assembler U Compiler ⊆ Linker ⊆ Loader ⊆ SystemSoftware (f) BatchOperatingSystem ⊆ RealTimeOperatingSystem ⊆ TypesOfOperatingSystem (g)
(2)	Using ≡ TYPE B	SecondaryMemory ≡ Memory ∩ isVolatile.false (a) ProcessIdentifier ≡ PID (b) ShortTermScheduler ≡ CpuScheduler (c)

Table 6 Sample of ABox axioms of OS ontology

S.No	Type	Axiom
(1)	Instance of a class	ApplicationPrograms (Spreadsheets) (a) NonPreemptiveSchedulingAlgorithm (FirstComeFirstServe) (b)
(2)	Relationship Type between two classes	isUsedBy(Hardware, ApplicationPrograms) (a) isTriggered(Interrupt, Software) (b) controlCoordinates(OperatingSystem, Hardware) (c) hasDirectAccess(CentralProcessingUnit, Memory) (d)
(3)	Datatype Property of an Instance	isVolatile(?Value)

'Hardware', 'OperatingSystem', and 'Interrupt', seen in Table 5 in 1a, 1b, 1c and 1d. Concepts of 'DirectAccess' and 'SequentialAccess' are sub-concepts under 'AccessMethods' as shown in Table 5 in 1e. Similarly, the same can be said for 1f and 1 g, as shown in Table 5. Concept 'SecondaryMemory', a nonvolatile part of Memory, shown in Table 5 in 2a. At the same time, 2b and 2c in Table 5 show that the concept 'ProcessIdentifier' is equivalently called 'PID' and 'ShortTermScheduler' is also known as 'CpuScheduler'.

The ABox axioms depict the instances of the concepts like, Spreadsheets are a kind of ApplicationPrograms while FirstComeFirstServe is an instance of NonPreemptiveSchedulingAlgorithm. This axiom is represented in Table 6 shown in 1a and 1b. While 2a in Table 6 represents object-type property 'isUsedBy' between Hardware and ApplicationPrograms stating that 'ApplicationPrograms use hardware. Object-type properties represented in 2b, 2c, and 2d shown in Table 6 are: 'isTriggeredBy', 'controlsCoordinates', and 'hasDirectAccess'. The datatype property' isVolatile(?Value') in 3 shown in Table 6 denotes the boolean value of the attribute 'isVolatile'.

Table 7 shows the sample of SWRL rules for the constructed ontology. In Tables 5 and 7, each strategy is given a reference tag (e.g., TYPE A, TYPE B, TYPE C, TYPE D, and TYPE E) to aid in understanding. TYPE A pertains to subset notation while TYPE B represents equivalence used in DL as shown in Table 5. TYPE C represents the datatype property in consequent denoting concept 'SecondaryMemory' with datatype property 'hasVolatile' having value False. TYPE D represents the datatype property in the antecedent, representing that 'The process in the main memory is in the ready state'. At the same time, TYPE E denotes the relationship or object property in antecedent used in SWRL in Table 7. This rule represents the constraint that 'The scheduler which selects the process from the secondary memory and loads it to the main memory is called the Long Term Scheduler'.

The rules and axioms in Tables 5, 6 and 7 are used to create the ITree and then generate Wh-type questions from the ontology in OBT as discussed in 4.2. There are a total of 126 rules, out of which 87 are SWRL, while the remaining 39 belong to DL, stating Tbox and Abox axioms, respectively. Out of 39 DL rules, 31 pertain to TYPE A, while the rest belong to TYPE B. Among the 87 SWRL rules, two belong to TYPE C, five cater to TYPE D, and the remaining 80 pertain to TYPE E. Each rule of the ontology generates one question, so with 126 rules, 126 questions have been generated from OBT.

Table 8 shows the classification of total number of Wh-type question stem generated based on the strategies. Each question stem generated has been classified concerning the type of question and whether it satisfies the required cognitive level based on Bloom's Taxonomy as shown in Tables 9. All generated Wh-type question stem satisfy the Level I and Level II cognitive level. The question type classification has been done based on Tables 1 and 2.

MBT The MBT uses a dataset with 2900 preprocessed sentences. Based on the manually generated Topic_list, the first stage of sentence selection selects 1100 sentences as input for key sentence identification. Specifically, in the second stage, different classification models, as shown in algorithm 2, are trained with Feature_set, and each classification model outputs the Key_sentences. The classifiers are trained by splitting the dataset into a 75–25% ratio. So with 1100 sentences in the sample dataset, 825 are split as training dataset while the remaining 275 is the testing dataset. The accurate classifier is the classifier that outputs the maximum number of Key_sentences. In this direction, SVM is the accurate classifier as SVM outputs the maximum number of Key_sentences compared to the other classification models shown in Table 10a. Table 10b shows the results of the SVM classifier using all criteria in terms of a confusion matrix.

Now, the system is programmed to generate three Cloze questions for one key sentence. So to reduce the overhead time complexity, only four sections of the one

Table 7 SWRL rules of OS ontology

S.No	Strategy	Rule
(1)	Using Datatype Property in the consequent TYPE C	SecondaryMemory(?j) → hasVolatile(?j, false)
(2)	Using Datatype Property in the antecedent TYPE D	stateOfProcess(?p) ∧ isIn(?p,?m) ∧ MainMemory(?m) → ReadyState(?p)
(3)	Using Relationship Type Property in the antecedent TYPE E	Scheduler(?s) ∧ selectsProcessFrom(?s,?d) ∧ SecondaryMemory(?d) ∧ loadsInto(?s,?m) ∧ MainMemory(?m) → LongTermScheduler

Table 8 Total number of questions from OBT

Rule	Number of rules	Number of questions generated	Based on
DL	39	31	TYPE A
		8	TYPE B
SWRL	87	2	TYPE C
		5	TYPE D
		80	TYPE E
Total Questions from Ontology		126	

chapter different than the original dataset are randomly chosen to generate Cloze questions for these sections. The system uses SVM as the accurate classifier and, with an input of 103 sentences, identifies 79 Key_sentences in the second stage. The keywords are identified based on the Word_prd of each word, and 3 Cloze questions are generated for each Key_sentence. Therefore in the third stage, MBT gives an output of 237 Cloze questions.

Figure 10 shows the sample of Cloze question stem and key generated in MBT as per the algorithm 2. The Cloze questions are categorized to cater to the recalling level based on Bloom's Taxonomy based on Tables 1 and 2. The proposed system through MBT generates 237 Cloze questions and with OBT generates 126 Wh-type questions, which makes a total of 363 questions from the chosen dataset.

Evaluation

In this section, the empirical evaluation is done to check the grammatical correctness and usefulness of the question items generated from the proposed system. Additionally, the evaluation of OBT and MBT algorithms is verified by generating questions across various technical courses.

Manual Evaluation

Five domain experts have manually evaluated all of the 363 generated questions. The evaluators are subject experts handling the course for the students in the college. The entire set of MCQ stem with the correct answer, or key is given to the evaluators. The guidelines to evaluate the set using a 3-point Likert scale is provided to the evaluators. Table 11 tabulates the evaluator's points for the questions.

The highest score on the Likert scale indicates the question satisfying the criteria, whereas the lowest score means vice versa. Each question in the question set is provided with check boxes and guidelines. The guidelines help the evaluator to evaluate the question based on the criteria and accordingly give the points.

Table 9 Sample of Wh-type Questions generated and classified based on Bloom's Taxonomy

Level	Strategy	Question generated	Type	No. of questions
Understand	TYPE A DL	WHICH AMONG THE FOLLOWING IS NOT A TYPE OF COOPERATING PROCESS? For rule 1 g shown in Table 5	Odd One Out	31
Remember	TYPE B DL	WHAT IS THE EQUIVALENT NAME OF A(N) CPUSCHEDULER? For rule 2c shown in Table 5	Synonym	8
Understand	TYPE C SWRL	WHAT IS THE VOLATILE VALUE FOR A(N) SECONDARY MEMORY? For rule 3 shown in Table 6	T/F	2
Remember	TYPE D SWRL	WHAT IS A(N) READY STATE? For rule 2 shown in Table 7	Definition	5
Understand	TYPE E SWRL	WHAT IS A(N) SCHEDULER THAT SELECTS PROCESS FROM A(N) SECONDARY MEMORY AND LOADES INTO A(N) MAIN MEMORY? For rule 3 shown in Table 7	One word answer	80

Table 10 MBT classifier's results

(a) Maximum No. of Key Sentences for each classifier	
Classifier	Max. No. of key sentences
LR	885
KNN	859
GNB	965
DT	832
SVM	992
(b) Results of the SVM Classifier using all features	
Metrics	Values
Accuracy	90.242
Precision	90.32
Recall	96.55
F-Measure	93.33

STEM	KEY
A. As a result of _______ scheduling, we can improve both the utilization of the CPU and the speed of the computer's response to its users	CPU
B. _______________ is normally disabled but will start if many processes are running and are using a threshold amount of memory	Swapping
C. __________ is allocated to processes until finally, the memory requirements of the next process cannot be satisfied that is, no available block of memory or hole is large enough to hold the next process	Memory
D. For example, if the base register holds 300040 and the limit register is _________, then the program can legally access all addresses from 300040 through 420939 (inclusive)	120900

Fig. 10 Sample of cloze questions generated

- **Useful:** Select the stem if it is useful in the assessment of the domain. Such questions are given 3 points
- **Useful with modified blank:** Applicable to only Cloze question stems. The generated Cloze question is selected only if the blank word is modified with a multi-

Table 11 3 point likert scale evaluation

Points	Usefulness	Grammar
3	Useful	Correct
2	Useful with modified blank	Major modification
1	Useless	Incorrect

word rather than a single word as chosen by the algorithm 2. Figure 11 shows an example based on this parameter. Such questions are given 2 points

– **Useless:** In the case of a Cloze question, select the stem if it has verbs chosen as blank, while in the case of a Wh-type question, select the stem if it is useless. Such questions are given 1 point

The Wh-type stems are also evaluated based on one additional parameter of grammar with the following guidelines:

– **Correct Grammar:** Select the stem if it is grammatically correct. Such questions are given 3 points
– **Major modifications in Grammar:** Select the stem if significant modifications are required in the grammar for the question, i.e., in cases with missing one or many connectives. Such questions are given 2 points
– **Incorrect Grammar:** Select the stem if it is grammatically incorrect in cases where the question structure does not provide any meaning. Such questions are given 1 point

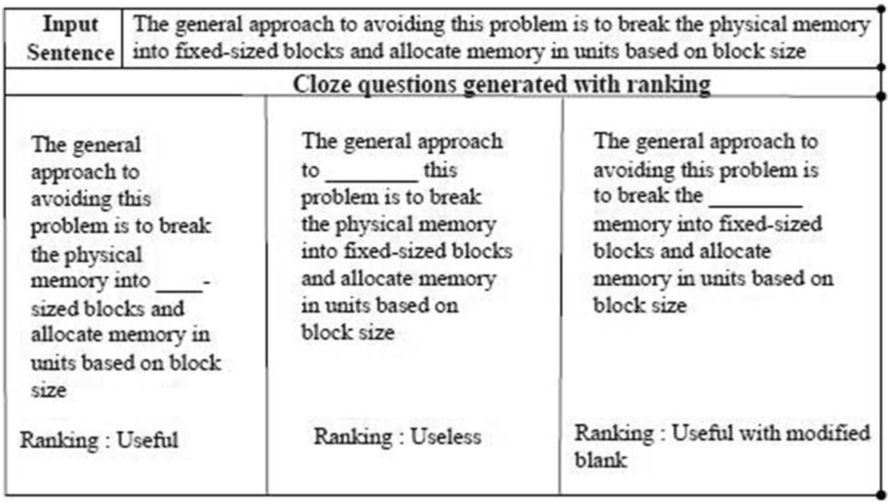


Fig. 11 Question ranking done by experts concerning Usefulness parameter

Wh-type questions are rated useless only when they cannot be administered for assessment. Furthermore, the Wh-type questions rated with significant modifications in grammar can be administered once after the subject expert modifies them. Therefore for OBT, the Wh-type questions rated with 3 points are only considered. In MBT, Cloze questions rated Useful with modified blank can be administered in the MCQ but will pertain to Level I cognitive level of Bloom's taxonomy. If modified to include two consecutive words substituted for blanks, these questions can cater to the Level II cognitive level of Bloom's taxonomy. Hence, in the MBT approach, Cloze questions rated with 3 and 2 points are considered useful. The reliability between the evaluators, is tested for the system using inter-rate reliability scores. Cohen's Kappa coefficients gives the inter-rater agreement scores across the evaluation done between the two evaluators for each metric (Cohen, 1968). In the proposed system, five evaluators are compared to each other for all questions using the same metric. Therefore, scores are computed for the ten combinations of pairs of evaluators for the usefulness and grammatical correctness criteria.

The coefficient for usefulness criteria is computed separately for the Cloze and Wh-type questions. Table 12 provides the weighted Cohen Kappa's coefficients for the experiment. As per the Cohen Kappa's Coefficient value, the evaluators moderately agree on the useful parameter for Wh-type questions. Substantial agreement on grammar parameters for Wh-type questions. For Cloze's questions, the evaluators were in substantial agreement on useful parameter.

Results

Table 13 shows the sample Cloze questions with their evaluation ratings. Based on the manual evaluation, Table 14 shows the number of sentences out of the total 237 Cloze questions that satisfied the overall usefulness metrics.

Table 15 shows the different questions generated from OBT and their evaluation metrics done by the experts. Manual evaluation of 126 Wh-type questions generated from OBT by five domain experts based on the metrics of Usefulness and Grammar is shown in Table 16. A question is identified as useful or grammatically correct only if at least three evaluators have categorized it. Based on this threshold, the number of questions satisfying the metric has been tabulated in Table 17. Here the useful parameter for Cloze questions considers all stems rated as useful and useful with modified blank as both are useful. However, the latter requires multi-words to be replaced by blanks.

Table 12 Weighted Cohen Kappa's coefficients

Parameter	Type of questions	Kappa's coefficients
Useful	Wh-type	0.598
Grammar		0.612
Useful	Cloze	0.827

Table 13 Sample of Cloze questions with evaluators' ratings

Stem	Key	Rating
________ is normally disabled but will selected if many processes are running and are using threshold amount of memory	Swapping	Useful
Although seemingly simple in concept _____ memory exhibits perhaps the widest range of type, technology, organization, performance and cost of any features of a computer system	computer	Useful with modified blank (computer memory)
Further the _____ unit portion of the processor may also require its own internal memory	control	Useful with modified blank (control unit)
In this, when a partition is free, a process is _____ from the input queue and is loaded into the free partition	selected	Useless
Swapping is normally disabled but will ____ if many processes are running and are using threshold amount of memory	start	Useless

Table 14 Manual evaluation of 237 Cloze questions

Evaluators	Parameters		
	Useful	Useful with modified blank	Useless
Ev1	114	34	89
Ev2	120	30	87
Ev3	117	32	88
Ev4	115	33	89
Ev5	117	32	88

As seen in Table 17, the manual evaluation shows that both Cloze and Wh-type questions generated by our proposed approach are promising in terms of usefulness and grammatical correctness and can be used to assess the courses in the technical domain.

MCQs Generation with Different Datasets

The application of the approach is checked in a different context by generating questions in OBT by taking two ontologies available online. One ontology is from a non-technical domain with SWRL rules, while the other is a pre-existing ontology built on a course from a technical domain without SWRL rules. The ontology having SWRL rules is Family Health History ontology (Peace, 2009) downloaded from the bioportal site(one of the repositories of ontologies), representing the family members' health history conditions. The SWRL rules here comprise two rules, one stating biological relationships of three generations based on parentage and another family history of health based on personal health findings. There are about 160 rules, more than 400 classes, one datatype property, and more than 150 object-type properties. Table 18 shows the rules in the ontology and the questions generated for each of them in the proposed approach.

As seen in Table 18 the generated questions query on the relationships based on the SWRL. The questions are general questions pertaining to the SWRL rules added to the ontology. Moreover the nomenclature in the ontology concepts has also given us grammatical correct questions. One another ontology taken for a technical domain is Data Structure and Algorithms (DSA) ontology a preliminary ontology built and used for this comparison.This ontology is used to manually add 10 SWRL rules along with 6 DL rules. With the manually created rules, Table 19 shows the sample questions for the DSA ontology generated from the proposed approach. Based on Table 19 it can be seen that the added rules have generated all the five different types of Wh-questions. The nomenclature in the ontology is similar to the proposed approach, thereby giving grammatically correct questions. Faculty experts evaluated all 16 questions for the course and rated them as useful to be administered in real-time.

For verifying the algorithm of MBT, a Computer Organization course is used, from which the Cloze questions are generated. The PDF file of the course is

Table 15 Sample of Wh-type questions with evaluators' ratings

Rule	Question generated	Rating for usefulness	Rating for grammar
AdvisoryLock≡MandatoryLock	WHAT IS THE EQUIVALENT NAME OF ADVISORY LOCK?	Useful	Correct Grammar
CpuUtilization U ResponseTime U Throughput U WaitingTime U TAT ⊆ CpuSchedulingCriteria	WHICH AMONG THE FOLLOWING IS NOT A TYPE OF CPU SCHEDULING CRITERIA?	Useful	Correct Grammar
Problem(?e) ∧ Process(?p) ∧ isContinouslyDenied(?p,?r) ∧ Resources(?r) → Starvation(?e)	WHAT IS THE PROBLEM WHEN A PROCESS IS CONTINOUSLY DENIED RESOURCES?	Useful	Correct Grammar
DirectoryOperation(?s) ∧ searches(?s,?d) ∧ DirectoryStructure(?d) ∧ toFindentryForAParticular(?s,?f) ∧ File(?f) → SearchForFile(?s)	WHAT IS A(N) DIRECTORY OPERATION THAT SEARCHES AND TO FIND ENTRY FOR A PARTICULAR A(N) FILE?	Useful	Incorrect Grammar
SecondaryMemory(?u) ∧ Mainmeory(?u1) ∧ hasCost(?u,?value) ∧ hasCost(?u1,?value1) → greaterThan(?u,?u1)	WHAT IS THE RELATIONSHIP WHEN THERE IS A(N) SECONDARY MEMORY THAT COST AN A(N) MAIN MEMORY THAT HAS COST?	Useless	Major Modifications
AccessMethods(?p) ∧ hasSequenceAccess(?p,true) → SequentialAccess(?p)	WHAT IS A(N) ACCESS METHODS THAT HAS SEQUENTIAL ACCESS EQUAL AS TRUE	Useless	Correct Grammar

Table 16 Manual evaluation of 126 Wh-type questions

Evaluators	Parameters				
	Useful	Useless	Correct grammar	Major modifications in grammar	Incorrect grammar
Ev1	120	6	116	5	5
Ev2	121	5	118	4	4
Ev3	120	6	115	6	5
Ev4	122	4	116	6	4
Ev5	122	4	115	7	4

Table 17 Summary of results for both techniques

Parameters	MBT	OBT	Total Questions	Percentage
Usefulness	149	121	270	74.38%
Grammar	237	116	353	97.24%

Table 18 Sample of Rules and Questions generated from the Family Health History Ontology

Rules and its equivalent generated Questions from Family History Ontology
has_natural_father(?a,?f), has_natural_mother(?a,?m), has_natural_father(?b,?f), has_natural_mother(?b,?n), DifferentFrom(?m,?n), Person_Female(?b)→has_half_sister(?a,?b)
WHAT IS THE RELATIONSHIP WHEN THERE IS HAS NATURAL FATHER AND HAS NATURAL MOTHER AND A(N) PERSON FEMAILE THAT HAS NATURAL FATHER IT AND HAS NATURAL MOTHER?
has_half_aunt(?a,?b), Person_Female(?a)→has_half_niece(?b,?a)
WHAT IS THE RELATIONAHSIP WHEN THERE IS AND A(N) PERSON FEMALE THAT HAS HALF AUNT?
has_mathernal_maternal_grandaunt(?a,?b), has_health_state(?b,?c)→has_fam_hx_in_maternal_maternal_grandaunt(?a,?c)
WHAT IS THE RELATIONSHIP WHEN THERE IS HAS MATERNAL MATERNAL GRANDAUNT HAS HEALTH STATE?_

preprocessed, from which the key sentences are identified by SVM that are then transformed into Cloze questions. From an input of 131 sentences, 98 were identified as key sentences, generating 294 Cloze questions. Figure 12 shows the different Cloze questions generated from the Computer Organization course through MBT.

An evaluation of about ten generated Cloze questions by the subject expert showed that the Cloze questions are also useful in real-time applications. Based on this observation, the proposed system can be adapted to any course in the technical domain to generate Cloze and Wh-type questions, which can satisfy the Level I and Level II cognitive skills based on Bloom's Taxonomy. Moreover, the system can generate questions from pre-existing ontology, provided certain modifications are made to the naming convention used for the ontology components.

Table 19 Sample of questions generated from DSA ontology

Strategy	Rules and questions
DL	*DICTIONARY ⊆ HASH_MAP U HASH_TABLE* WHICH AMONG THE FOLLOWING IS NOT A TYPE OF DICTIONARY?
DL	*MATRIX_MULTIPLICATION_ALGORITHM ⊆ coppersmithWinogradAlgorithm U strassenAlogithm* WHICH AMONG THE FOLLOWING IS NOT A TYPE OF MATRIX MULTIPLICATION ALGORITHM?
SWRL	*LIST(?L), has_node(?L,?N),SINGLY_LINKED_LIST_NODE(?N) → HEAP_OPERATION(?O)* WHAT IS A(N) GRAPH OPERATION THAT OPERATES ON A(N) HEAP?
SWRL	*OPERATION(?O), operates_on(?O,?A), DISJOINT_SET(?A) → DISJOINT_SET_OPERATION(?O)* WHAT IS A(N) OPERATION THAT OPERATES ON A(N) DISJOINT SET?

STEM	KEY
A. Thus, a _____ retrieved based on a portion of its contents rather than its address.	word
B. Tape units, discussed in are _____ access.	sequential
C. As with ordinary random-access memory, each location has its own addressing mechanism and retrieval time is ________ independent of location or prior access patterns.	constant
D. For internal memory, ______ is typically expressed in terms of bytes (1 byte or 8 bits) or words.	capacity

Fig. 12 Sample of Cloze questions generated from "Computer Organization" ebook

Comparison of the Proposed Approach with Existing Approaches

Compared with Tables 3 and 4, the proposed approach has attempted to generate MCQs in the technical domain. The research generates five types of MCQs stems that are better in relevance and grammatical correctness. Furthermore, the questions generated in this research pertain to either Level I or Level II cognitive criteria. Moreover, the proposed approach is a unique attempt to combine both techniques because of their advantages in generating different questions. In addition, the questions generated in the proposed research can also be administered in real-time scenarios based on human evaluation. Table 20 shows the comparison of the proposed approach with the existing approaches listed in 3 and 4.

Conclusion

The e-learning facilitates the usage of MCQ as an assessment tool for the concepts learned by the examinees. MCQ can be quickly evaluated and administered to a vast set of examinees at a given time. In addition, MCQ has become a very popular questionnaire in the current e-learning systems because MCQ can assess the different cognitive skills of the examinees. However, the manual construction of MCQ is quite cumbersome and error-prone. As a result, the automatic generation of such questionnaires has been popular in recent decades to gain the benefits of MCQ. Existing efforts need to generate such a questionnaire for a technical domain. Since semantic-based approaches apply semantics, these approaches can generate grammatically correct Wh-type questions automatically compared to ML approaches. In comparison, ML approaches use the same input data and perform better than semantic-based approaches to generate Cloze questions automatically. Moreover, the existing researchers still need to evaluate the questionnaire to assess cognitive skills using Bloom's Taxonomy.

This research proposes a hybrid method using OBT and MBT to generate MCQ stems automatically, which can assess cognitive skills based on Bloom's Taxonomy. The novel framework using OBT and MBT can generate grammatically correct Cloze questions compared to existing works that utilize only ML technique. The proposed approach demonstrates that using DL and SWRL rules in an ontology has generated five different Wh-type question stems. The proposed approach has been evaluated on a synthetic ontology primarily built for the experiment along with few other ontologies. The experiments prove that the approach is capable of generating MCQ stems that are both useful and grammatically correct, as evaluated by the domain experts. An empirical study also generates Cloze and Wh-type questions from a real-time dataset and hence it can be adapted to assess students.

The proposed approach using OBT generates Wh-type questions comprising of a single concept on the consequent which could also be experimented with multiple concepts on the consequent. While using MBT, the framework generates reasonably good Cloze questions, except for a few cases where a multi-word need to be replaced by a blank to make the Cloze question more useful. This research has only targeted the generation of MCQ stems with their corresponding key. To use these MCQ stems, distractors need to be generated which is an exciting task to pursue in future research. In addition, there is also a scope for aligning the questions to the course objectives. Furthermore, the limitation observed in this research is the initial time required to create an ontology.

Table 20 Comparison of the proposed approach with existing approaches listed in Tables 3 and 4

Question type	Bloom's taxonomy	Strategy	Remarks
Wh-type, T/F, Definition. Synonyms, Odd one-out using Ontology	Recall and Understanding	Uses SWRL and DL Rules to generate questions from ontology	Majority of the questions are relevant and can be used in real-time applications
Cloze using Text	Recall	Uses topics to select initial set of sentences, then features of count (nouns), presence of abbreviations, appendix words and sentence length used to generate Cloze questions from text	

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

References

Agarwal, M., & Mannem, P. (2011). Automatic gap-fill question generation from text books. In Proceedings of the sixth workshop on innovative use of NLP for building educational applications, pages 56–64.

Agarwal, M. (2012). Cloze and open cloze question generation systems and their evaluation guidelines. International Institute of Information Technology, Hyderabad.

Aldabe, I., Maritxalar, M., & Mitkov, R. (2009). A study on the automatic selection of candidate sentences distractors. In AIED, pages 656–658.

Alsubait, T., Parsia, B., & Sattler, U. (2012). Mining ontologies for analogy questions: A similarity-based approach. In OWLED, volume 849.

Alsubait, T. (2015). Ontology-based question generation. PhD thesis, University of Manchester.

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

Baader, F., Horrocks, I., & Sattler, U. (2005). Description logics as ontology languages for the semantic web. In Mechanizing mathematical reasoning, pages 228–248. Springer.

Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals. Cognitive domain.

Brown, J., Frishkoff, G., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 819–826.

Carneson, J., Delpierre, G., & Masters, K. (1996). Designing and managing multiple choice questions. Retrieved March, 30:2008.

Ch, D. R., & Saha, S. K. (2018). Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies, 13*(1), 14–25.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213.

Considine, J., Botti, M., & Thomas, S. (2005). Design, format, validity and reliability of multiple choice questions for use in nursing research and education. *Collegian, 12*(1), 19–24.

Correia, R., Baptista, J., Eskenazi, M., & Mamede, N. (2012). Automatic generation of cloze question stems. In International Conference on Computational Processing of the Portuguese Language, pages 168–178. Springer.

Cubric, M., & Tosic, M. (2011). Towards automatic generation of e-assessment using semantic web technologies. *International Journal of e-Assessment*.

D'Sa, J. L., & Visbal-Dionaldo, M. L. (2017). Analysis of multiple choice questions: Item difficulty, discrimination index and distractor efficiency. *International Journal of Nursing Education*, *9*(3).
Das, B., & Majumder, M. (2017). Factual open cloze question generation for assessment of learner's knowledge. *International Journal of Educational Technology in Higher Education, 14*(1), 1–12.
Dunham, B., Yapa, G., & Yu, E. (2015). Calibrating the difficulty of an assessment tool: The blooming of a statistics examination. *Journal of Statistics Education*, *23*(3).
Effenberger, T. (2015). Automatic question generation and adaptive practice. PhD thesis, Masarykova univerzita, Fakulta informatiky.
Eiter, T., Ianni, G., Krennwallner, T., & Polleres, A. (2008). Rules and ontologies for the semantic web. In *Reasoning web* (pages 1–53). Springer.
Faizan, A., & Lohmann, S. (2018). Automatic generation of multiple choice questions from slide content using linked data. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, pages 1–8.
Gierl, M. J., Bulut, O., Guo, Q., & Zhang, X. (2017). Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of Educational Research, 87*(6), 1082–1116.
Grosof, B. N., Horrocks, I., Volz, R., & Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In Proceedings of the 12th international conference on World Wide Web, pages 48–57.
Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199–220.
Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies, 43*(5–6), 907–928.
Haladyna, T. M., & Rodriguez, M. C. (2013). *Developing and validating test items*. Routledge.
Hansen, J. D., & Dexter, L. (1997). Quality multiple-choice test questions: Item-writing guidelines and an analysis of auditing testbanks. *Journal of Education for Business, 73*(2), 94–97.
Heilman, M. (2011). Automatic factual question generation from text. PhD thesis, Carnegie Mellon University.
Holohan, E., Melia, M., McMullen, D., & Pahl, C. (2005). *Adaptive e-learning content generation based on semantic web technology*.
Horrocks, I. (2005). Owl: A description logic based ontology language. In *International conference on principles and practice of constraint programming* (pages 5–8). Springer.
Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission, 21*(79), 1–31.
Jelenkovi, F., & TOˇSI, M. (2015). Semantic multiple-choice question generation and concept-based assessment filip jelenkoviˆu and milorad toˇsiˆu. Vistas of English for Specific Purposes, page 325.
Krathwohl, D. R. (2002). A revision of bloom's taxonomy: An overview. *Theory into Practice, 41*(4), 212–218.
Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education, 30*(1), 121–204.
Le, N. T., Kojiri, T., & Pinkwart, N. (2014). Automatic question generation for educational applications–the state of art. In *Advanced Computational Methods for Knowledge Engineering: Proceedings of the 2nd International Conference on Computer Science, Applied Mathematics and Applications* (ICCSAMA 2014) (pp. 325–338). Springer International Publishing.
Leo, J., Kurdi, G., Matentzoglu, N., Parsia, B., Sattler, U., Forge, S., Donato, G., & Dowling, W. (2019). Ontology-based generation of medical, multiterm mcqs. *International Journal of Artificial Intelligence in Education, 29*(2), 145–188.
Majumder, M., & Saha, S. K. (2015). A system for generating multiple choice questions: With a novel approach for sentence selection. In Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications, pages 64–72.
Mostow, J., & Chen, W. (2009). Generating instruction automatically for the reading strategy of self-questioning. In AIED, pages 465–472.
Mostow, J., & Jang, H. (2012). Generating diagnostic multiple choice comprehension cloze questions. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 136–146.
Narayanan, S., Adithan, M., et al. (2015). Analysis of question papers in engineering courses with respect to hots (higher order thinking skills). *American Journal of Engineering Education (AJEE), 6*(1), 1–10.

O'Dwyer, A. (2012). Experiences of assessment using multiple-choice questions on advanced modules taken by level 8 and level 9 engineering students. *AISHE-J: The All Ireland Journal of Teaching and Learning in Higher Education*, *4*(1).

Palmer, E. J., & Devitt, P. G. (2007). Assessment of higher order cognitive skills in undergraduate education: Modified essay or multiple choice questions? Research paper. *BMC Medical Education, 7*(1), 1–7.

Papasalouros, A., Kanaris, K., & Kotis, K. (2008). Automatic generation of multiple choice questions from domain ontologies. In e-Learning, pages 427–434. Citeseer.

Parsia, B., Alsubait, T., Leo, J., Malais´e, V., Forge, S., Gregory, M., & Allen, A. (2015). Lifting emmet to owl getting the most from skos. In International Experiences and Directions Workshop on OWL, pages 69–80. Springer.

Peace, J. (2009). Family Health History Ontology. Bio-Portal. https://bioportal.bioontology.org/ontologies/FHHO

Pino, J., Heilman, M., & Eskenazi, M. (2008). A selection strategy to improve cloze question quality. In Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada, pages 22–32.

Rakangor, S., & Ghodasara, Y. (2015). Literature review of automatic question generation systems. *International Journal of Scientific and Research Publications, 5*(1), 1–5.

Rus, V., Cai, Z., & Graesser, A. (2008). Question generation: Example of a multi-year evaluation campaign. Proc WS on the QGSTEC.

Sahathanavijayan, A., Balasundaram, S., Narayanan, S. H., Kumar, S. V., & Prasad, V. V. (2017). Automatic generation of multiple choice questions for e-assessment. *International Journal of Signal and Imaging Systems Engineering, 10*(1–2), 54–62.

Shah, R., Shah, D., & Kurup, L. (2017). Automatic question generation for intelligent tutoring systems. In 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), pages 127–132. IEEE.

Silberschatch, A., Galvin, P. B., & Gagne, G. (2006). *Operating System Principles* (7th ed.). Wiley.

Stanford Center for Biomedical Research. (2019). *Prot´eg´e- a free, open-source ontology editor and framework for building intelligent systems*.

Stasaski, K., & Hearst, M. A. (2017). Multiple choice question generation utilizing an ontology. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 303–312.

Tarrant, M., Knierim, A., Hayes, S. K., & Ware, J. (2006). The frequency of item writing flaws in multiple-choice questions used in high stakes nursing assessments. *Nurse Education Today, 26*(8), 662–671.

Tarrant, M., & Ware, J. (2012). A framework for improving the quality of multiple-choice assessments. *Nurse Educator, 37*(3), 98–104.

Testa, S., Toscano, A., & Rosato, R. (2018). Distractor efficiency in an item pool for a statistics classroom exam: Assessing its relation with item cognitive level classified according to bloom's taxonomy. *Frontiers in Psychology, 9*, 1585.

Venugopal, V. E., Alsubait, T., & Kumar, P. S. (2016). Modeling of item difficulty for ontology-based mcqs. arXiv preprint arXiv:1607.00869.

Venugopal, E. V., & Kumar, P. S. (2015). A novel approach to generate mcqs from domain ontology: Considering dl semantics and open-world assumption. *Journal of Web Semantics, 34*, 40–54.

Vinu, E. V., & Kumar, P. S. (2015). Improving large-scale assessment tests by ontology based approach. In The Twenty-Eighth International Flairs Conference.

Wood, T. J. (2009). The effect of reused questions on repeat examinees. *Advances in Health Sciences Education, 14*(4), 465–473.

Xie, J., Peng, N., Cai, Y., Wang, T., & Huang, Q. (2022). Diverse distractor generation for constructing high-quality multiple choice questions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30*, 280–291.

Yaneva, V., et al. (2018). Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval. In Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, pages 389–398.

Zoumpatianos, K., Papasalouros, A., & Kotis, K. (2011). Automated transformation of swrl rules into multiple-choice questions. In Twenty-Fourth International FLAIRS Conference.

Authors and Affiliations

Archana Praveen Kumar[1] · **Ashalatha Nayak**[1] · **Manjula Shenoy K**[2] · **Chaitanya**[1] · **Kaustav Ghosh**[1]

Archana Praveen Kumar
archana.kumar@manipal.edu

Manjula Shenoy K
manju.shenoy@manipal.edu

Chaitanya
chaitanya.mitma@learner.manipal.edu

Kaustav Ghosh
kaustav.ghosh1@learner.manipal.edu

1 Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India 576104

2 Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India 576104