2024 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES) | 979-8-3503-7613-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SPICES62143.2024.10779773

Generation of Bloom's taxonomy-based complex-level questions using knowledge graph

Blessy Paul P
Research Scholar
School of Computer Sciences, MG University,
Kottayam, Kerala, India
blessypaul91@gmail.com

Cini Kurian
Principal
Al-Ameen College, Edathala
Kerala, India
cinikurian@gmail.com

***Abstract*—In the landscape of educational technology, the present scenario in question generation often involves static and predefined assessments that may not adequately gauge students' diverse cognitive abilities. The relevance of a knowledge graph in this context lies in its ability to capture the inherent connections and hierarchies within a domain, fostering a more intelligent and context-aware question generation process. This research proposes a novel approach to question generation, leveraging the knowledge graph. The knowledge graph serves as a dynamic representation of interconnected concepts, offering a structured framework for understanding relationships between topics. This, in turn, enables a more nuanced and contextually relevant selection of keywords for question formulation. Unlike traditional question generation methods, this model harnesses the power of semantic relationships encoded in the knowledge graph to align questions with Bloom's Taxonomy levels, ensuring a spectrum of complexity. This innovative approach not only modernizes question generation but also aligns assessments more closely with real-world problem-solving scenarios, ultimately contributing to a more effective and personalized learning experience.**

***Index Terms*—Natural Language Processing (NLP), Bloom's Taxonomy, Question Generation, New Education Policy, Knowledge Graph**

I. Introduction

In the dynamic field of education, the current system for question generation has evolved to encompass a sophisticated blend of cognitive theories, pedagogical strategies, and cutting-edge technologies. At the core of this evolution lies the enduring relevance of Bloom's Taxonomy - a seminal framework introduced by Benjamin Bloom in 1956 (Aricvitch et al. 2020) [1]. As per the explanation of Pikhart et al. (2019) [2], this hierarchical model classifies cognitive skills into six distinct levels: knowledge (remember), comprehension (understand), application (apply), analysis (Analyse), synthesis (evaluate), and generation(create). Its significance lies in providing educators with a structured guide for formulating questions that align with specific cognitive processes, fostering the progressive development of critical thinking skills in learners.

The process of generating questions based on Bloom's Taxonomy begins with foundational inquiries aimed at knowledge and comprehension, gradually ascending to more complex cognitive skills. This structured approach ensures that assessments are not merely focused on recall but also challenge students to apply, analyze, synthesize, and evaluate information (Kanwal et al. 2020) [3]. As students ascend the cognitive hierarchy, they engage in a deeper and more nuanced understanding of the subject matter. This alignment of question generation with Bloom's Taxonomy serves as a pedagogical compass, steering educators towards assessments that go beyond surface-level learning and encourage the cultivation of higher-order thinking skills.

In parallel, the integration of knowledge graphs has emerged as a transformative force in question generation, particularly at the higher levels of Bloom's Taxonomy. Knowledge graphs are graphical representations that depict the intricate relationships between various entities and concepts. By leveraging the interconnected nature of knowledge, educators can craft questions that transcend isolated facts, prompting students to synthesize information across diverse domains (Setyowati et al. 2022) [4]. For instance, a question might require the analysis of historical events in conjunction with scientific principles, fostering a holistic understanding of complex topics. This approach not only mirrors real-world problem-solving scenarios but also enriches assessments by contextualizing knowledge within a broader framework.The relevance of knowledge graphs extends beyond their capacity to create complex-level questions. They also offer a personalized dimension to assessments. By tapping into a vast web of information and understanding the individual learner's profile, educators can tailor questions that challenge and stimulate intellectual growth (Wijanarko et al. 2021) [5]. This personalized approach not only enhances the effectiveness of assessments but also promotes inclusivity by accommodating diverse learning styles and preferences. As technology continues to advance, the fusion of Bloom's Taxonomy with knowledge graphs promises to redefine the boundaries of question generation, providing educators with powerful tools to create engaging and effective assessments that align with the ever-evolving needs of learners in the 21st century.

The aim of the research is to create more sophisticated and contextually rich educational assessments. The motivation lies in enhancing critical thinking and deeper learning by leveraging the structured relationships in knowledge graphs.

This research seeks to address drawbacks such as bias, scalability issues, and a lack of interpretability in existing models. By resolving these, the paper aims to improve the accuracy, relevance, and fairness of the generated questions, ultimately fostering a more effective and transparent assessment process. This research study creates subtopics by using the syllabus as an input. The relationship between subjects and their corresponding subtopics can be highlighted by creating a knowledge graph with nodes representing the subtopics. Following the mapping of the subtopics and Bloom's taxonomy levels, many complex-level questions are generated.

II. Literature Review

In Natural Language Processing (NLP), the generation of questions has become a focal point, with researchers employing advanced models to automate this process and enhance various applications such as educational technology and information retrieval systems. A seminal work by Du et al. (2017) [6] delves into neural network-based question generation, demonstrating the effectiveness of their model in extracting relevant information from passages. The model incorporates attention mechanisms, allowing it to focus on key elements within the context, thereby improving the quality and contextual relevance of the generated questions.

The utilization of sequence-to-sequence learning has been a significant trend in question generation models. Serban et al. (2016) [7] explored this approach by employing recurrent neural networks to capture the sequential dependencies in textual data. Their model showcased a capacity to generate fluent and coherent questions by understanding the intricate relationships between words in a given context. This highlights the importance of capturing contextual nuances for effective question generation.

The Transformer architecture, introduced by Vaswani et al. (2017) [8], has also left an indelible mark on NLP applications, including question generation. The self-attention mechanism inherent in Transformers enables models to efficiently process and weigh contextual information across a sequence of words. This mechanism contributes to the creation of complex questions that are not only grammatically correct but also contextually informed, reflecting a more nuanced understanding of the input data.

Furthermore, recent advancements in question generation have witnessed the integration of trained large language models, such as BERT (Devlin et al., 2018) [9] and GPT (Brown et al., 2020) [10]. These models leverage large-scale pre-training on diverse language data, allowing them to capture intricate linguistic patterns and contextual nuances. Fine-tuning such models for specific question generation tasks has proven successful in achieving state-of-the-art results, showcasing the potential of transfer learning in NLP-based question generation.

The integration of knowledge graphs in question generation has emerged as a pivotal area of research, offering a structured framework for contextualized and sophisticated inquiries. Models such as KPEQA (Shen et al., 2019) [11]leverage knowledge graph embeddings to enhance question generation by incorporating relational information. This approach ensures that questions are formulated with a deeper understanding of the relationships between entities, contributing to contextually rich inquiries.

Moreover, the work by Zhang et al. (2020) [12] introduces a knowledge-aware question generation model that effectively exploits information from knowledge graphs. By incorporating graph attention networks, the model attends to relevant entities and relationships, improving the precision and relevance of generated questions.The study conducted by Wang et al. (2019) [13] explores a knowledge graph-based approach for generating multiple-choice questions. Their model leverages a graph convolution network to capture architecture from knowledge graphs, demonstrating enhanced performance in generating diverse and contextually aligned questions. In a different vein, the work of Lin et al. (2020) [14] introduces an innovative approach that leverages reinforcement learning and knowledge graphs for question generation. This model dynamically adjusts question generation based on the relevance and importance of entities within the knowledge graph, showcasing adaptability in diverse contexts.

Despite the significant contributions of knowledge graph-based question generation models, there are notable drawbacks that warrant attention. Firstly, these models are susceptible to biases inherent in the underlying knowledge graphs, potentially leading to the generation of biased questions. Scalability is another concern, as the vast and complex nature of knowledge graphs can impede efficient processing, resulting in incomplete contextual understanding. Additionally, issues of interpretability arise, as the decision-making process of these models may be opaque, hindering user trust. The models may also struggle with out-of-domain entities and fail to adapt to dynamic knowledge landscapes. Finally, the static nature of knowledge graphs poses challenges in capturing temporal dynamics, impacting the relevance and accuracy of generated questions. Addressing these limitations is crucial for ensuring the fairness, adaptability, and transparency of knowledge graph-based question generation models in diverse applications.

In this study, the drawbacks have been considered, and new model for automatic question generation has been developed. This research paper accepts the syllabus as input and generates different complex-level of questions by leveraging knowledge graphs.

III. Data and Pre-Processing

The proposed system involves two main aspects of data: the knowledge graph representing educational topics and their relationships, and the Bloom's Taxonomy levels with associated verbs. The knowledge graph is structured as a dictionary, where each key represents a broad educational topic, and the associated values are lists of subtopics within that topic. This hierarchical structure forms the basis of the knowledge graph, capturing relationships between different educational concepts. For example, 'Programming' may include subtopics such as

TABLE I
BLOOM'S TAXONOMY LEVELS

Taxonomy level	Question words	Question types
Knowledge	define, who, when, where, quote, name,identify,label	Remembering facts, Recall events or features
Comprehension	Differentiate, distinguish, describe, summarize, discuss, predict, list, contrast	Understanding, Compare, Interpret
Application	Demonstrate, calculate, solve, illustrate, examine, test, classify	Represent real life application, Problem solving
Analysis	Analyse, explain, classify, connect, identify, compare	Analyse pattern, Recognition,Comparison
Evaluation	Access, rank, grade, support, conclude, measure, select	Choose, Verify evidence, Recognize, Assess theories
Creativity	Design, invent, compose, create, hypothesise	Creative thinking, Innovative ideas

'Variables,' 'Loops,' and 'Functions.' Similarly, Bloom's Taxonomy levels are organized into a dictionary, with cognitive levels ('Remember,' 'Understand,' etc.) as keys and associated verbs as values. Table 1 Shows different taxonomy levels.

In terms of data preprocessing, the knowledge graph structure inherently organizes topics and subtopics. No explicit preprocessing steps are required for this simple representation. However, if the knowledge graph data were sourced from external files or databases, additional preprocessing might be necessary to handle any inconsistencies, format the data appropriately, or remove irrelevant information. For the generated questions, the program employs string formatting to construct question templates based on the randomly selected topic, subtopic, and Bloom's level. This step ensures the creation of coherent and grammatically correct complex-level questions. Overall, the program's data and preprocessing steps are streamlined, allowing for a flexible and dynamic generation of questions based on the provided educational framework. Following are the libraries used for data preprocessing.

A. Spacy

Spacy is a mobile NLP library designed to handle and interpret vast amounts of text, making it a favourite among enthusiasts. It includes a variety of models regarding entities, vocabularies, syntaxes, and trained vectors. (Zouaq, A. et al. 2021) [15]. The requirements will determine how these models are loaded. 'en_core_web_sm' is the default package for the "english-core-web" package, where "sm" stands for small. The English language contains three models of Spacy: small, medium, and large.

```
import spacy
spacy.load (“en_core_web_lg”)
```

B. Natural Language Toolkit

All NLP libraries are derived from NLTK, or the Natural Language Processing Toolkit. It offers lexical resources, more than 50 corpora, and a collection of libraries for a variety of tasks, including tokenization, stemming, classification, tagging, and semantic reasoning. It's a Python platform for developing applications that need natural language processing (Joshi, S. et al. 2021) [16]. A key element of the subtopic generating method described in this work is NLTK.

1) Lancaster Stemmer: In NLP, stemming is the reduction of words to the corresponding root or stem. It's possible that this word stem and a dictionary-based root term are not the same. Yet, it is merely a truncated or equivalent version of the term. For example, "acceptance," "accepted," and "acceptance" all stem from the word "accept." Lancaster Stemmer is used in the coding during question development to stem the verb from the syllabus and generate a skeleton of the question after POS tags the sentence from which the complicated question is to be constructed. This algorithm can occasionally turn words into weird roots, thus, it's important to pay attention to spelling mistakes in sentences when using it.The dictionary says that "programming" is not a stem word in the example below; however, the stemmer turns words like "programmi" into the stem "program." Lancaster = LancasterStemmer()

```
print(“Python : ”, Lancaster.stem(“Python”))
print(“program : ”, Lancaster.stem(“program”))
print(“programming : ”, Lancaster.stem(“programming”))
Output:
Python : Python
program : program
programming : program
```

2) WordNet Lemmatizer: Lemmatization is the process of grouping many word forms together and analysing them as one entity to determine the root word (from dictionary), which is preferably referred to as a "lemma." Lemmatization and stemming are somewhat comparable.

An extensive lexical database of the English language is given to the public for free and is called WordNet. Comparable terms are arranged into sets (synsets) in what might be thought of as a thesaurus, with each word expressing a unique idea on its own. The primary goal is to establish an organised semantic relationship between words. An interface for using the WordNet corpus reader, a dictionary, is offered by NLTK. To lemmatize words, a WordNetLemmatizer() instance is required upon download and installation, much like in the stemming example.

```
lemmatizer = WordNetLemmatizer()
print(“program : ”, lemmatizer.lemmatize(“program”))
print(“programs : ”, lemmatizer.lemmatize(“programs”))
print(“corpora : ”, lemmatizer.lemmatize(“corpora”))
Output:
program : program
programs : program
corpora : corpus
```

3) Part-of-Speech(POS) Tagging: The process of assigning the Part-of-Speech label to a token within a text_corpus is known as POS tagging. One of the most potent features of the NLTK is the POS tagger. After reading the sentence, it gives each token a part of speech (noun, verb, adjective, etc.).

TABLE II
List of POS Tags

Part of Speech	Tags	Example
Noun:Singular	NN	Computer
Noun:Plural	NNS	Computers
Proper Noun:Singular	NNP	John
Proper Noun:Plural	NNPS	Indians
Verb:Base form	VB	Eat
Verb:Past tense	VBD	Ate
Verb:Past participle	VBN	Eaten
Verb:Present	VBP	Are
Verb:Present participle	VBG	Talking
Personal Pronoun	PRP	I,He,She
Wh-Adverb	WRB	What, When, Why

Each segment of speech has its own tag. We provide special attention to NN, NNS, NNP, NNPS, VB, VBN, VBD, PRP, and PRPP during the question creation process. Table 2 explains each tag and corresponding meaning.

Example:

import nltk print(nltk.pos_tag (nltk.word_tokenize(”Hey, what are you talking?”)))

Output:

[(’Hey’, ’NNP’), (’,’, ’,’), (’what’, ’WRB’), (’are’, ’VBP’), (’you’, ’PRP’), (’talking’, ’VBG’), (’?’, ’.’)]

IV. Methodology

The methodology for generating Bloom’s taxonomy-based complex-level questions using knowledge graphs involves several key steps, illustrated in the figure 1, below. Initially, the given syllabus is parsed to extract key concepts and topics. These extracted elements are then mapped onto a pre-existing knowledge graph, which encapsulates a vast web of interconnected information. Using the knowledge graph, relevant entities and their relationships are identified and contextualized within the framework of Bloom’s Taxonomy. The model employs natural language processing (NLP) techniques to analyze these relationships and generate questions aligned with higher-order cognitive skills, such as analysis, synthesis, and evaluation. A question generation algorithm, integrated with graph attention networks, focuses on pertinent nodes and edges to formulate contextually rich questions. Finally, these questions are refined through a post-processing step to ensure grammatical correctness and alignment with educational standards. This systematic approach ensures that the generated questions are both relevant and challenging, promoting deeper understanding and critical thinking.

This pipeline ensures a structured, scalable, and pedagogically sound method for generating complex-level questions. The individual responsible for creating the curriculum must submit a paragraph-long piece of writing as input. The supplied input is divided into discrete pieces known as tokens. The NLTK Punkt Sentence Tokenizer, which divides the running text into coherent sentences, can be used for tokenization. The initial stages of pre-processing incoming data also involve lemmatization, stemming, and lowercasing. WordNet Lemmatizer is used for lemmatization, and the Lancaster Stemmer algorithm is used for stemming.

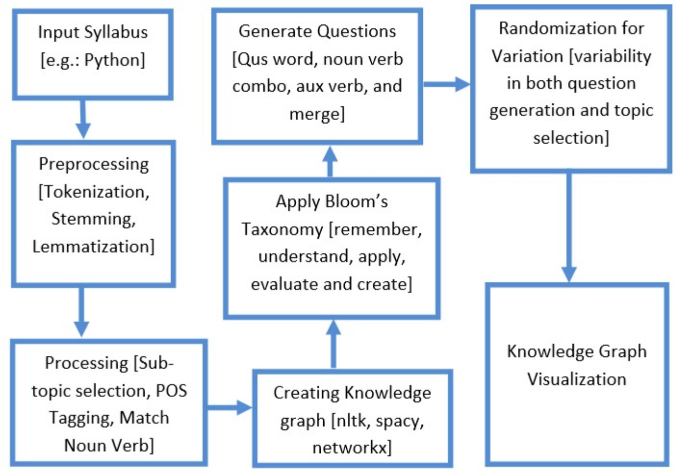

Fig. 1. Structure of proposed system

In the k-level representation of Bloom’s Taxonomy within the architecture, each level (k) corresponds to a distinct cognitive process: remember, understand, apply, analyze, evaluate, and create. The system maps syllabus concepts to these levels, ensuring questions are generated to target specific cognitive skills. For instance, lower levels focus on recall and comprehension, while higher levels involve synthesis and critical evaluation, guiding the question generation algorithm to create appropriately challenging questions. Discourse cues are the basis for choosing possible sentences. Sentences containing one of seven hardcoded discourse markers are separated from the other sentences. After determining the discourse marker in a sentence, the mapping to the possible question form is completed. After POS tagging, the sentence is divided into the subject topic part and the question-word part. Auxiliary verbs are located by going through sentences. If the auxiliary verb is present, the question can be formed by tokenizing the portion of the sentence that will include the auxiliary verb, separating it from the main verb, and then using the auxiliary verb to make the question. The phrase string is joined once more, but with one modification: a wh-question at the beginning and a question mark at the conclusion take the place of the auxiliary verb. Instead of wh-question words, Bloom’s taxonomy-level-based words also used. Figure 2 shows a classification knowledge graph as per the inputed syllabus based on the topic ”python program”. The graph has been generated using python library ’matplotlib’.

V. Implementation and Results

This automatic question generation system combines knowledge graph representation, Bloom’s Taxonomy, and graph visualization. The methodology can be elucidated in the following steps:

A. Knowledge Graph Representation

The foundation of this system is the representation of a knowledge graph, a structured set of interconnected educational topics and their corresponding subtopics. The knowl-

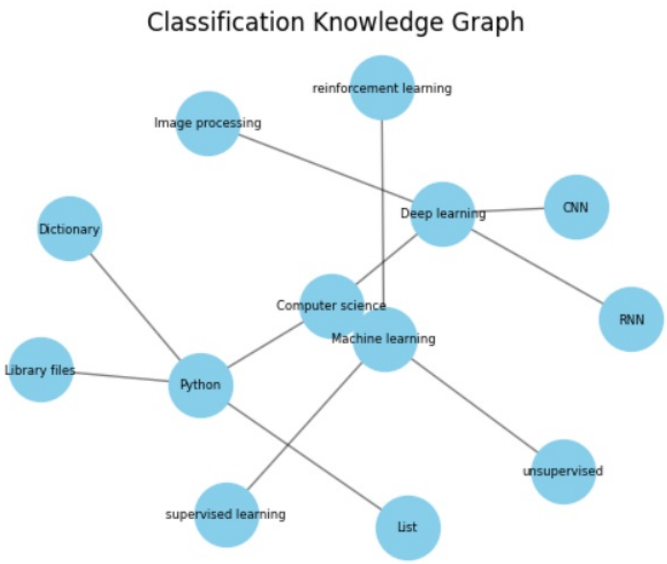

Fig. 2. Classification Knowledge graph

edge graph is implemented as a dictionary in Python, where each key signifies a broad topic (e.g., 'Programming,' 'Mathematics') and the associated values are lists of subtopics within that topic (e.g., 'Variables,' 'Loops'). This hierarchical structure captures the relationships between different educational concepts and serves as the basis for both question generation and graph visualization.

```
knowledge_graph = {
'computer science': ['Programming', 'Machine Learning', 'Operating System'],
'Programming': ['Variables', 'Loops', 'Functions'],
'Machine Learning': ['ANN', 'LR', 'Decision Tree'],
'Operating System': ['Memory management', 'File Management', 'IO Management'] }
```

B. Bloom's Taxonomy for Question Generation

The code incorporates Bloom's Taxonomy, a cognitive framework that categorizes educational objectives into six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. Each level is associated with a set of verbs indicative of the cognitive skills expected at that level. Random selection is employed to choose a broad topic from the knowledge graph and a Bloom's Taxonomy level. Subsequently, the program randomly selects a subtopic within the chosen topic and a verb associated with the selected Bloom's level. This randomization ensures variability in the generated questions, allowing for a diverse set of queries that span different cognitive domains.

```
# Bloom's Taxonomy levels
bloom_levels = { 'Remember': ['Define', 'List', 'Recall'],
'Understand': ['Explain', 'Describe', 'Summarize'],
'Apply': ['Apply', 'Demonstrate', 'Use'],
'Analyze': ['Analyze', 'Compare', 'Contrast'],
'Evaluate': ['Evaluate', 'Assess', 'Judge'],
'Create': ['Create', 'Design', 'Generate'] }
```

C. Question Generation and String Formatting

The generate_complex_question function combines the selected topic, subtopic, and verb to construct a complex-level question template. Using string formatting, the function creates a grammatically correct and coherent question, such as "What is the recall of Variables in the field of Programming?" This process enables the generation of contextually relevant questions that assess specific cognitive skills within the chosen educational context. The dynamic nature of the question generation ensures that each run of the program produces a unique question based on the random selections.

```
def generate_complex_question(topic, bloom_level):
    subtopic = random.choice(knowledge_graph.get(topic, []))
    verb = random.choice(bloom_levels[bloom_level])
    question_template = f"verb.lower() subtopic in the field of topic?"
    return question_template
```

D. Knowledge Graph Visualization

To provide a visual representation of the educational relationships within the knowledge graph, the program employs the NetworkX library for graph creation and Matplotlib for visualization. The draw_knowledge_graph function transforms the knowledge graph into a graph structure, where nodes represent both broad topics and subtopics, and edges connect them, denoting relationships. The resulting graph is displayed using Matplotlib, allowing users to visually explore the hierarchical structure of educational concepts. This visualization enhances the understanding of how different topics and subtopics are interconnected.

```
def draw_knowledge_graph(graph):
    G = nx.Graph()
    for topic, subtopics in graph.items():
        for subtopic in subtopics:
            G.add_edge(topic, subtopic)
    pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, font_weight='bold', node_color='skyblue', node_size=1000, font_size=8, edge_color='black')
    plt.show()
```

E. Program Execution and Randomization

The main function orchestrates the program's execution. It begins by randomly selecting a topic and a Bloom's Taxonomy level, ensuring that each run introduces variation. After generating a complex-level question using the selected components, the program proceeds to draw and display the knowledge graph. The combination of randomization, question generation, and graph visualization provides users with an interactive and dynamic tool for exploring educational content. Figure 3 shows the output generated when the input word is given as "memory management" from the syllabus Operating System. Figure 4 shows the result generated after randomization, for the input tocken "LR" (Logistic Regression) from the syllabus machine Learning.

```
def main():
    # Choose a random topic from the knowledge graph
    topic = random.choice(list(knowledge_graph.keys()))
    # Choose a random Bloom's Taxonomy level
    bloom_level = random.choice(list(bloom_levels.keys()))
    # Generate a complex-level question
    complex_question = generate_complex_question(topic,
bloom_level)
    # Draw the knowledge graph and print question
    draw_knowledge_graph(knowledge_graph)
    print(complex_question)
```

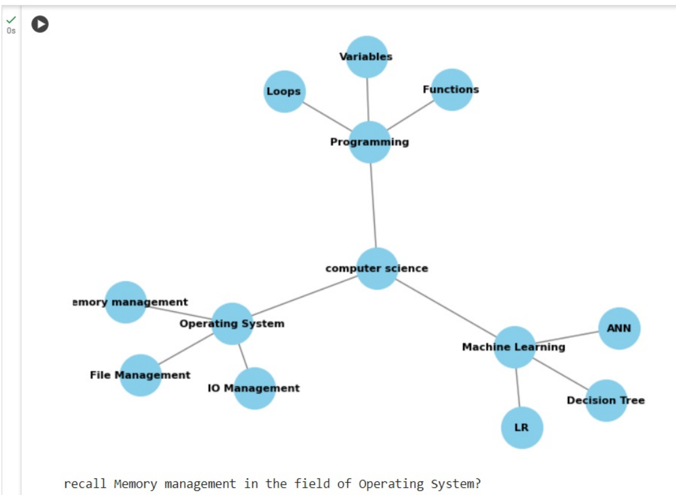

Fig. 3. Auto-generated question and corresponding knowledge graph

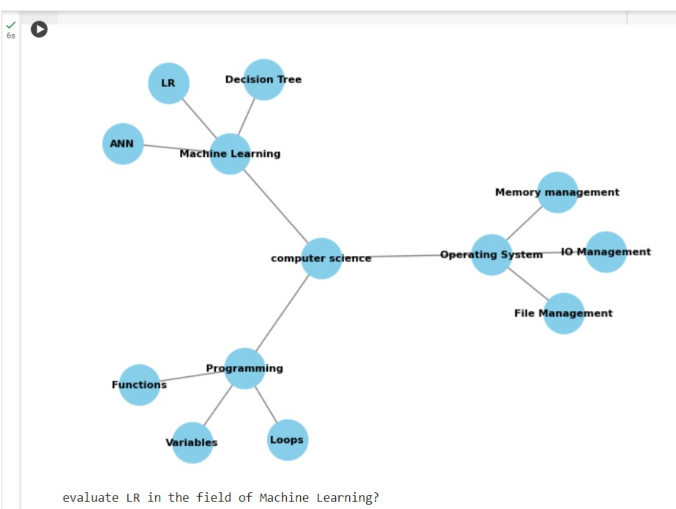

Fig. 4. Auto-generated question after randomization

VI. Conclusion

In the present scenario, where personalized and adaptive learning experiences are gaining prominence, the ability to dynamically generate questions aligns with the demand for tailored assessments. This approach acknowledges the diversity of learners' strengths and provides a nuanced evaluation of their cognitive skills. In conclusion, the generation of Bloom's Taxonomy-based complex-level questions using a knowledge graph presents a dynamic and contextually relevant approach to educational assessment. The major results of this endeavor include the creation of diverse and intricate questions that span various cognitive domains, ensuring a comprehensive evaluation of learners' understanding. By leveraging randomization, the program generates unique questions for each run, fostering adaptability and catering to the evolving needs of educational assessment.

References

[1] Arievitch, I. M. (2020). Reprint of: The vision of Developmental Teaching and Learning and Bloom's Taxonomy of educational objectives. *Learning, culture and social interaction*, 27, 100473.
[2] Pikhart, M.,& Klimova, B. (2019). Utilization of linguistic aspects of Bloom's taxonomy in blended learning. *Education Sciences*, 9(3), 235.
[3] Gul, R., Kanwal, S.,& Khan, S. S. (2020). Preferences of the teachers in employing revised blooms taxonomy in their instructions. sjesr, 3(2), 258-266.
[4] Setyowati, Y., Susanto, S.,& Munir, A. (2022). Critical Thinking within the Context of the Revised Bloom's Taxonomy in Written Language Tests. *Budapest International Research and Critics Institute-Journal (BIRCI-Journal)*, 5(2).
[5] Wijanarko, B. D., Heryadi, Y., Toba, H.,& Budiharto, W. (2021). Question generation model based on key-phrase, context-free grammar, and Bloom's taxonomy. *Education and Information Technologies*, 26, 2207-2223.
[6] Du, X.,& Cardie, C. (2017, September). Identifying where to focus in reading comprehension for neural question generation. In Proceedings of the 2017 conference on *empirical methods in natural language processing* (pp. 2067-2073).
[7] Serban, I. V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A.,& Bengio, Y. (2016). Generating factoid questions with recurrent neural networks: *The 30m factoid question-answer corpus*. arXiv preprint arXiv:1603.06807.
[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.,& Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
[9] Alberti, C., Andor, D., Pitler, E., Devlin, J.,& Collins, M. (2019). Synthetic QA corpora generation with roundtrip consistency. *arXiv preprint* arXiv:1906.05416.
[10] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P.,& Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.
[11] Xu, X., Wang, T., Yang, Y., Hanjalic, A.,& Shen, H. T. (2020). Radial graph convolutional network for visual question generation. *IEEE transactions on neural networks and learning systems*, 32(4), 1654-1667.
[12] Qiu, Y., Wang, Y., Jin, X.,& Zhang, K. (2020, January). Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *Proceedings of the 13th international conference on web search and data mining* (pp. 474-482).
[13] Xiong, W., Yu, M., Chang, S., Guo, X.,& Wang, W. Y. (2019). Improving question answering over incomplete kbs with knowledge-aware reader. *arXiv preprint* arXiv:1905.07098.
[14] Feng, Y., Chen, X., Lin, B. Y., Wang, P., Yan, J.,& Ren, X. (2020). Scalable multi-hop relational reasoning for knowledge-aware question answering. *arXiv preprint arXiv:2005*.00646.
[15] Martel, F.,& Zouaq, A. (2021, March). Taxonomy extraction using knowledge graph embeddings and hierarchical clustering. *In Proceedings of the 36th Annual ACM Symposium on Applied Computing* (pp. 836-844).
[16] Joshi, S., Shah, P., & Shah, S. (2021). Automatic question paper generation, according to bloom's taxonomy, by generating questions from text using natural language processing. International Research Journal of Engineering and Technology, 6, 495-501.