machine learning & knowledge extraction

MDPI

Article

Generating Multiple-Choice Knowledge Questions with Interpretable Difficulty Estimation Using Knowledge Graphs and Large Language Models

Mehmet Can Şakiroğlu [1], Halil Altay Güvenir [1] and Kamer Kaya [2,3,*]

[1] Computer Engineering Department, Bilkent University, Ankara 06800, Türkiye; can.sakiroglu@bilkent.edu.tr (M.C.Ş.); guvenir@cs.bilkent.edu.tr (H.A.G.)
[2] Faculty of Engineering and Natural Sciences, Sabanci University, İstanbul 34956, Türkiye
[3] Center of Excellence in Data Analytics (VERIM), Sabanci University, İstanbul 34956, Türkiye
* Correspondence: kaya@sabanciuniv.edu

Abstract

Generating multiple-choice questions (MCQs) with difficulty estimation remains challenging in automated MCQ-generation systems used in adaptive, AI-assisted education. This study proposes a novel methodology for generating MCQs with difficulty estimation from input documents by utilizing knowledge graphs (KGs) and large language models (LLMs). Our approach uses an LLM to construct a KG from input documents, from which MCQs are then systematically generated. Each MCQ is generated by selecting a node from the KG as the key, sampling a related triple or quintuple—optionally augmented with an extra triple—and prompting an LLM to generate a corresponding stem from these graph components. Distractors are then selected from the KG. For each MCQ, nine difficulty signals are computed and combined into a unified difficulty score using a data-driven approach. Within a 150-MCQ, proof-of-concept dataset from Wikipedia, the proposed signals show interpretable associations with empirical incorrect-answer rates aligning with human responses/performance. The results support the feasibility of the proposed pipeline, yet a larger-scale human study may be required to establish deployment-scale validity. Our approach improves automated MCQ generation by integrating structured knowledge representations with LLMs and a data-driven difficulty estimation model.

Keywords: multiple-choice question generation; difficulty estimation; interpretability; knowledge graph; large language models

Academic Editor: Ján Paralič

Received: 3 April 2026
Revised: 7 May 2026
Accepted: 15 May 2026
Published: 20 May 2026

1. Introduction

Multiple-choice questions are a common form of assessment used in educational settings due to their utility in standardized testing, self-assessment, and adaptive learning systems. A multiple-choice question typically consists of a question or statement known as the "*stem*" followed by several possible answers. One of these answers is the correct one, referred to as the "*key*", while the others are incorrect, known as "*distractors*". An example multiple-choice question whose parts are annotated can be seen in Figure 1. The effectiveness of a multiple-choice question depends on the quality of its stem, key, and distractors, as they must be clear, concise, and appropriately challenging for the intended audience. Automated multiple-choice question generation offers the potential to reduce the time and effort required for manual question generation in education while maintaining quality and relevance.

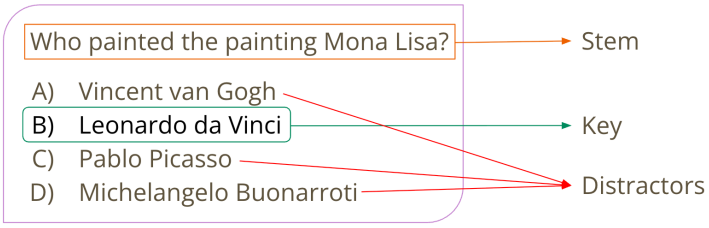

Figure 1. An example multiple-choice question.

Even though multiple-choice questions are well-known and widely used, automatically generating relevant and diverse multiple-choice knowledge questions with realistic difficulty estimation remains a challenging task. Existing methods either rely on handcrafted rules, which lack scalability, or black-box deep-learning approaches, which struggle with interpretability. An effective MCQ-generation system must not only generate relevant questions but also provide an interpretable difficulty estimation aligned with human response patterns. Furthermore, another challenge lies in the cost of creating an appropriate dataset, as it requires extensive human annotation.

To address such challenges, we propose a novel approach that combines knowledge graphs (KGs) with large language models (LLMs) for generating multiple-choice questions with interpretable difficulty estimation. The integration of LLMs enables advanced natural language understanding and generation capabilities, while the knowledge graph serves as a structured backbone for representing factual information and entity relationships. An illustrative example with a small knowledge graph is presented in Figure 2, where entities, their types, and semantic relationships are encoded as nodes and edges.

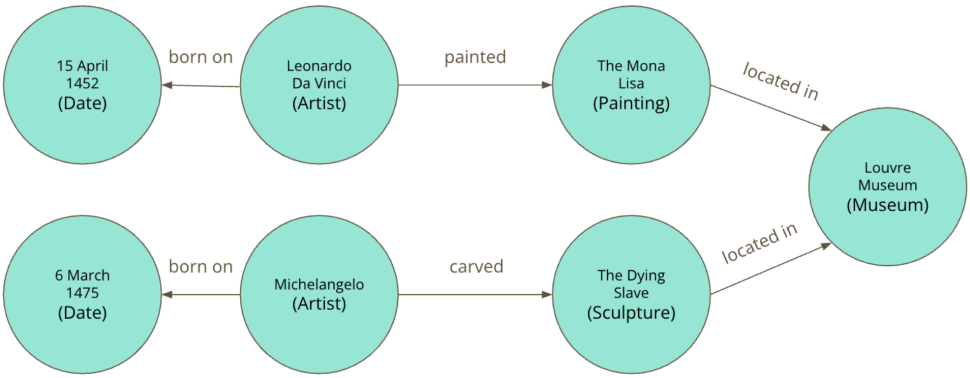

Figure 2. An illustrative example of a knowledge graph representing entities, their types, and the semantic relationships between them.

Our method begins by building a knowledge graph (KG) from the input text using an LLM. Multiple-choice questions (MCQs) are then systematically generated from this graph. For each MCQ, a key node serving as the correct answer is selected, along with a relevant triple or quintuple from the KG, which may be supplemented with an extra triple for additional context. These structured graph elements are then converted into a question stem by the LLM, and suitable distractors are also selected from the KG. To ensure explainability in difficulty estimation, we compute several intuitive and interpretable difficulty signals, including graph- and network-analytics-based metrics, embedding-based semantic similarities, and linguistic features, and combine them into a unified difficulty score in a data-driven manner by leveraging the human evaluation data we collect. Furthermore, beyond its technical aspects, our framework has substantial potential for enhancing educational practices, as it can be integrated into intelligent tutoring systems, adaptive testing platforms, and digital learning environments to deliver personalized assessments. Overall, the novel contributions of this work are summarized below:

- We propose a novel end-to-end framework for generating MCQs from textual sources. The framework constructs a knowledge graph from the input text, generates MCQs from the resulting graph, and estimates their difficulty in an interpretable manner. Our hybrid approach combines the structural knowledge of KGs with the natural language generation capabilities of LLMs to produce validated candidate MCQs.
- We construct a novel, yet small, dataset comprising MCQs, their corresponding knowledge subgraphs, and empirically derived difficulty labels based on an average of 38 independent human responses per question. To the best of our knowledge, this is the first dataset that jointly includes MCQs, their underlying knowledge subgraphs, and empirical difficulty labels derived from human evaluation.
- We introduce nine interpretable, data-driven difficulty estimation signals spanning graph topology, textual semantics, and linguistic complexity, thereby capturing both structural and semantic properties of the generated MCQs. Their effectiveness is assessed through four regression models and an ablation study.
- Finally, the proposed study is a successful initial validation for MCQ generation with difficulty estimation. Accordingly, the contribution is to demonstrate feasibility, identify interpretable difficulty signals, and expose methodological constraints that must be addressed before domain-specific or deployment-scale use.
- Our data and code are available to enable reproduction and further research (https://zenodo.org/records/20067934, accessed on 19 May 2026).

The remainder of this paper is organized as follows: Section 2 discusses related work in question generation, MCQ generation, difficulty modeling, and the use of knowledge graphs and LLMs in these settings. Section 3 details our methodology, including knowledge graph construction, multiple-choice question generation, and difficulty estimation. Section 4 details the dataset construction process, including the human evaluation protocol used to obtain empirical difficulty labels, and presents key statistics characterizing the resulting dataset. Section 5 presents the experimental results and analysis, and Section 6 discusses them. Finally, Section 7 concludes the paper, points out key limitations, and outlines directions for future research accordingly.

2. Background and Related Work

Automated question generation has seen advancements with various approaches exploring different aspects of the problem. A review of related works on this topic, including their improvement aspects and limitations, is conducted with comparisons to our own.

2.1. Graph- and Ontology-Based Question Generation

Reddy et al. [1] propose a method that employs recurrent neural networks (RNNs) rather than templates, as in their prior work, for question verbalization, thereby enhancing robustness and reducing the need for manual effort. However, the authors do not discuss the difficulty level of the question. Instead, they directly focus on the question text (stem) generation with the correct answer (key).

Elsahar et al. [2] present an encoder-decoder architecture that leverages textual contexts and a copy mechanism to address the challenge of the inability to generate questions for unseen predicates and entity types of previous works, hence enabling zero-shot question generation. Yet, Elsahar et al. restrict their task to generating natural language questions given a single input triple instead of multiple triples. They also do not emphasize parameters such as difficulty. Later, Chen et al. [3] propose a bidirectional Graph2Seq (Graph-to-Sequence) model to encode the KG subgraph to take advantage of the information provided by the graph structure. Then, they use an RNN decoder with a node-level copying mechanism to generate the question based on the output of a GNN-based (Graph

Neural Network) encoder they propose. However, Chen et al. do not control or investigate the difficulty of the generated questions.

Li et al. [4] take advantage of a pre-trained LLM, GPT-2, to obtain a comprehensive semantic context, and then they construct a graph using the entities identified within this context. Then, they utilize an answer-aware graph attention network (GAT) to update it based on the constructed graph to generate the question. However, the constructed graph is not a KG and its edges do not represent semantic relationships; instead, they represent the co-occurrence of two entities in the same paragraph. The authors also do not discuss the difficulty of the generated question.

Graph- and Ontology-Based Approaches Taking Difficulty into Account

Alsubait et al. [5] generate difficulty-controllable multiple-choice questions by utilizing ontologies to measure the similarity between the distractors and the key, based on the number of common properties they share in an ontology. They handle the stem generation with a template-based approach based on the ontology. Seyler et al. [6] provide a KG-based question generation technique that first selects an entity from the KG and then generates a SPARQL (SPARQL Protocol and RDF Query Language) query that specifies that node uniquely and converts the query to a natural language question with preset templates referred to as *question verbalization*. For difficulty estimation, their approach requires a question-answer corpus with annotated difficulties to train a classifier on. They propose generating distractors by relaxing their original query's constraints to retrieve more than one unique answer. The effect of the distractors on the question's difficulty is defined with the *confusion* metric computed via the difficulty classifier trained on the Q&A corpus.

Kumar et al. [7] study the generating complex, multi-hop questions that require reasoning across multiple KG triples, hence accepting subgraphs and multiple triples as its input, as opposed to a single triple. They present a Transformer-based model that also takes the difficulty into account. This work is similar to ours as it shares the goal of enabling the generation of multi-hop questions with controllable difficulty levels. However, Kumar et al. define difficulty exclusively based on two NER-dependent factors: (1) confidence scores for entity-mention linking and (2) the selectivity of entity surface forms, without incorporating structural or semantic properties of the question or the subgraph as a whole. Kusuma et al. [8] propose an ontology-based question generation framework that introduces a combination of taxonomy ontology and sentence ontology, referred to as knowledge ontology. Given a textual input, their system constructs an ontology and generates various types of questions using query templates. Both the type and difficulty of each question are determined manually by ontology engineers and domain experts. While their approach supports multiple question formats and provides difficulty annotations, it relies on template-based generation and does not incorporate automated difficulty estimation.

Bi et al. [9] introduce DiffQG, a difficulty-controllable single-answer question generation model with no distractors that produces natural language questions from a given KG subgraph with a specified difficulty level. Their approach uses a mixture-of-experts module to learn soft templates for different difficulty levels, enhancing the diversity of question phrasing without relying on manually crafted templates. They also incorporate a disentanglement module that isolates the KG triples relevant to the target difficulty, which enables counterfactual reasoning—training the model on perturbed subgraphs to reinforce a causal link between the difficulty label and the question features. Additionally, the authors propose a difficulty estimation mechanism called AutoDE that considers eight difficulty signals and the use of those signals by normalizing and linearly combining them. We follow a similar approach to estimate the difficulty with different signals, yet the assumption of linearity can be misleading, and to solve that, we propose a data-driven

methodology while employing both linear and non-linear models. Moreover, they propose the counterfactual reasoning approach to further enable the generation of questions of different difficulty levels given the same input subgraph, with the underlying idea that the given subgraph should not dictate the difficulty of the question. On the contrary, we think the topology of the given subgraph and the information within should be the main determining factor for difficulty, especially in knowledge questions. The broader challenge of text-based question difficulty prediction is systematically reviewed by AlKhuzaey et al. [10], who highlight the major role of linguistic features and the need for standardized datasets to enable meaningful comparisons between models.

Zhu et al. [11] quantify difficulty as the average similarity of each distractor to the correct answer. In their system, an MCQ can be generated in multiple versions: by selecting distractors that are very similar to the answer, the question's difficulty score increases, whereas using more dissimilar distractors yields an easier question. They implement an algorithm to automatically pick a set of distractors such that the resulting question's difficulty falls into a specified range or level. If impossible, they automatically lower the targeted level by one. To improve the accuracy of this difficulty metric, the authors refine the similarity calculation with topological weights, effectively incorporating knowledge from the structure of a semantic network to better judge how "close" a distractor is to the answer. This approach allows for explicit difficulty control: the same question stem can be paired with easier or harder sets of options, producing tiered versions of the question. One limitation is that this method predominantly captures only one aspect of difficulty: distractor ambiguity. It does not account for other factors like the intrinsic complexity of the question's content. Additionally, while the difficulty calculation is interpretable, they do not investigate a comprehensive list of signals that might affect the difficulty of an MCQ.

Wei and Hao [12] introduce KGNN-ADP, a KG-enhanced neural network designed for predicting the difficulty of an MCQ. Their framework leverages a domain-specific KG to extract relevant knowledge points for a given question and assess difficulty along two primary axes: (1) knowledge difficulty, computed by analyzing the semantic proximity and structural weight of the matched knowledge points in the graph, and (2) option difficulty, modeled using both the semantic similarity between distractors and the answer and the internal semantic divergence within the full question. While this work shares our emphasis on difficulty estimation grounded in KGs, it differs from ours in scope: their model predicts difficulty for existing MCQs rather than generating new ones by solely focusing on the absolute difficulty of MCQs, and not on the broader task of MCQ generation.

The prior research also extends to specialized topics. For example, an ontology-based approach by Leo et al. [13] has been applied in the medical field to generate multi-term, case-based multiple-choice questions (MCQs) from a medical ontology, which shows the value of structured knowledge in niche areas. Also, a hybrid framework by Kumar et al. [14] combines machine-learning and semantic techniques to create different types of MCQ stems for technical fields, which are then assessed against the cognitive levels of Bloom's Taxonomy. Additionally, the synergy between knowledge graphs and large language models has recently gained attention for related tasks as well, such as question answering, where combining structured KGs with LLMs improves performance [15,16].

2.2. Non-Graph-Based Approaches

Vachev et al. [17] present *Leaf*, a system for the end-to-end generation of MCQs from educational texts. The goal is to ease quiz/exam creation for instructors by inputting course material and outputting factual MCQs. The system uses neural models to perform the subtasks of question and distractor generation: it fine-tunes a Transformer-based text-to-text

model (T5) on Q&A pairs and trains another model to produce realistic distractors. Given a passage and a target answer, the former produces a question that becomes the input of the latter. This system demonstrates that large-scale pre-trained models can be harnessed to automate MCQ creation from unstructured text, achieving good fluency and relevance in both questions and distractors. *Leaf*, however, does not incorporate question difficulty; all the questions are intended to be high-quality, but there is no control over or estimation of the difficulty. Furthermore, *Leaf* relies on raw text input and does not utilize KGs; it may struggle with the coverage of specific relationships or while providing explanations. On the contrary, our work aims to estimate and explain the difficulty of each generated question.

Zeng et al. [18] propose RTRL, a deep question generation framework that combines a relation-aware transformer with reinforcement learning for deep question generation. RTRL introduces an encoder that embeds answer distances and words, and uses a BiLSTM to build answer-contextualized representations. RTRL also models both explicit linguistic relations and implicit semantic relations via its relation-aware transformer. The model is fine-tuned using reinforcement learning with sentence-level rewards to optimize evaluation metrics, thereby addressing the training-evaluation mismatch. Although RTRL significantly improves performance in deep question generation, it does not focus on multiple-choice question generation or difficulty modeling; its focus remains on deep question generation.

The research in this field has also moved beyond English-language applications, with works like Johnson et al. [19] demonstrating a parallel construction method to generate questions for Spanish textbooks by leveraging an existing English-based system.

Non-Graph-Based Approaches Taking Difficulty into Account

Gao et al. [20] explores difficulty control while generating open-ended questions for reading comprehension. They use text passages, where the input is a sentence from a reading comprehension article plus a target answer phrase, and the output is a question that asks about that answer at a specified difficulty level. A dataset with difficulty tags (over 70 k questions split into easy and hard), enabling the authors to train an end-to-end sequence-to-sequence question generation model with difficulty, is leveraged. The results show that their model can indeed tailor the generated questions to the requested difficulty without losing quality, as the questions remain fluent and answerable while matching the difficulty specification. Gao et al. thereby demonstrate the feasibility of controlling question difficulty. However, their work is limited to open-ended Q&A pairs in text and a binary difficulty distinction, since there is no provision for generating distractors as in MCQs. Cheng et al. [21] advances Gao et al. [20]'s approach by focusing on the reasoning complexity of questions. Noting that the prior method offered little interpretability regarding difficulty, they redefine difficulty as the number of inference hops or reasoning steps required to answer it. This strategy offers better interpretability and stronger logical controllability than treating difficulty as a latent label. However, the approach is applied to free-text Q&A and generates single-answer questions; it does not handle MCQs. Moreover, it restricts the definition of difficulty to a single parameter: the number of reasoning steps required.

2.3. A Brief Comparison with the Literature

Our work distinguishes itself from the literature by integrating easy-to-achieve structured knowledge with auto-generated knowledge graphs and large language models to generate validated candidate MCQs with interpretable difficulty estimation. Unlike prior QG work, we leverage a knowledge graph with an LLM, using the KG to ground the question in verifiable facts and the LLM to produce fluent questions by taking advantage of its pre-trained capabilities. This combination enables us to cover complex knowledge

that might require reasoning while maintaining natural language fluency and diversity by leveraging LLMs, a synergy not explored in previous systems to the best of our knowledge. Furthermore, our difficulty estimation system provides justifications for a question's difficulty rating based on the defined difficulty signals, offering interpretability. Lastly, we explicitly calibrate our difficulty predictions with the dataset we collect from human participants, ensuring that the difficulty levels correspond to the real world. This human-in-the-loop calibration means that our system's idea of difficulty is not just an assumption but reflects actual ease or struggle observed among the participants. Through this novel integration of KGs, LLMs, and a human-calibrated approach, our work delivers a more comprehensive solution for generating multiple-choice questions and estimating their difficulty. A comparison of the literature above, including this work, is presented in Table 1.

To clarify the contribution boundary, this work does not claim that KG-based QG or multi-signal difficulty modeling are new *in isolation*. The proposed pipeline connects text-derived KG construction, KG-subgraph-based MCQ stem generation, same-type distractor selection, interpretable graph/text/linguistic difficulty signals, and human-calibrated empirical difficulty labels in a single proof-of-concept MCQ workflow. Recent LLM-based MCQ studies also motivate a more careful validation framing as summarized in Table 2. Grévisse et al. [22] evaluate LLM-generated MCQs against item-writing guidelines and report issues such as ambiguous keys and implausible distractors, emphasizing the need for human oversight. Artsi et al. [23] systematically reviewed LLM-generated medical examinations and showed that validity evidence varies substantially across studies. For distractor generation, Feng et al. [24] showed that LLMs can produce mathematically valid distractors but may fail to capture common student misconceptions. For difficulty estimation, Zotos et al. [25] use LLM uncertainty as a predictor of MCQ difficulty, while our work instead focuses on interpretable KG-, text-, and linguistic signals derived during generation. The distractor-generation review by Awalurahman and Budi [26] further supports treating distractor plausibility and distractor evaluation as first-class validation criteria.

Table 1. Comparison of the presented literature, including the proposed study.

	Input	QG Model	Mul.-hop QG	MCQ Gen.	Difficulty Modeling	Human Calib. in Diff. Mod.
[1]	KG	RNN-based (seq2seq)	✗	✗	None	
[2]	KG triple and related texts	Encoder-decoder with copy mechanism	✗	✗	None	
[3]	KG	Bidirectional graph2seq model with copy mechanism	✓	✗	None	
[4]	Text/auto-generated entity co-occurrence graph	GPT-2, answer-aware GAT, and multi-head attention generation module	✓	✗	None	
[17]	Text	Transformer-based (T5)	✗	✓	None	
[18]	Text	Relation-aware transformer with reinforcement learning	✓	✗	None	
[5]	Ontology	Template-based	✗	✓	Difficulty control via key/distractor similarities based on number of common properties	✗
[6]	KG	SPARQL queries and templates	✗	✓	Difficulty estimation with logistic regression	✓(Ind.)

Table 1. *Cont.*

	Input	QG Model	Mul.-hop QG	MCQ Gen.	Difficulty Modeling	Human Calib. in Diff. Mod.
[7]	KG	Transformer-based	✓	✗	Difficulty control and estimation with custom formula	✗
[8]	Text, and its auto constructed ontology	SPARQL queries and templates	✓	✗	Manual via ontology engineers & domain experts	✓
[9]	KG	Soft-templates with MoE, and counterfactual reasoning	✓	✗	Difficulty control and estimation via linear modeling of proposed signals	✓
[11]	Semantic Network	Rule-based	✗	✓	Difficulty estimation and discrete control based on key/distractor similarities by also integrating topological weights	✗
[12]	Existing MCQ and KG	None			Difficulty prediction with the proposed KG-enhanced neural network based on Bi-LSTM	✗
[20]	Text	Seq2seq encoder-decoder based on LSTMs with copy mechanism	✗	✗	Difficulty control by initializing the hidden state of the decoder w.r.t. the required difficulty	✓(Ind.)
This work	Text/auto-gen. KG	Knowledge graph and LLM-based	✓	✓	Difficulty estimation with both linear and non-linear models based on signals	✓

Table 2. Validation evidence emphasized by recent LLM-based MCQ/question-generation and difficulty-estimation studies. A check mark indicates that the dimension is directly evaluated or central to the study; a dash indicates that it is absent or not a primary focus.

Study	Factual Correctness	Distractor Quality	Educational Validity	KG Quality	Human/Expert Validation
Grévisse et al. [22]	✓	✓	✓	–	domain-expert, author-side review
Artsi et al. [23]	✓	partial	✓	–	mixed across reviewed studies
Feng et al. [24]	✓mathematical validity	✓misconception alignment	✓	–	human evaluation
Zotos et al. [25]	–	–	✓difficulty via student correctness	–	student-response datasets
Bi et al. [9]	✓generated-question correctness partial	–	✓difficulty controllability	KG/subgraph central, not source audit	human evaluation
This work	automated validation plus author-side source-grounded screening	same-type KG distractors, LLM validation, and MCQ screening	human incorrect-answer calibration	author-side KG source-grounding audit	human response data plus author-side screening

3. Materials and Methods

In this section, we present the methodology adopted in our study, which consists of three main components: (1) KG construction, (2) MCQ generation, and (3) difficulty estimation. Figure 3 provides an overview of the proposed framework. Given a corpus of factual information, we first construct a structured KG that captures the entities and their semantic relationships. The KG serves as the foundation for generating candidate MCQs by selecting appropriate subgraphs and transforming them into MCQs. Finally, we estimate the difficulty of each MCQ using a combination of graph-based, embedding-based semantic similarities, and linguistic features. Each stage of the framework is elaborated in the following subsections.

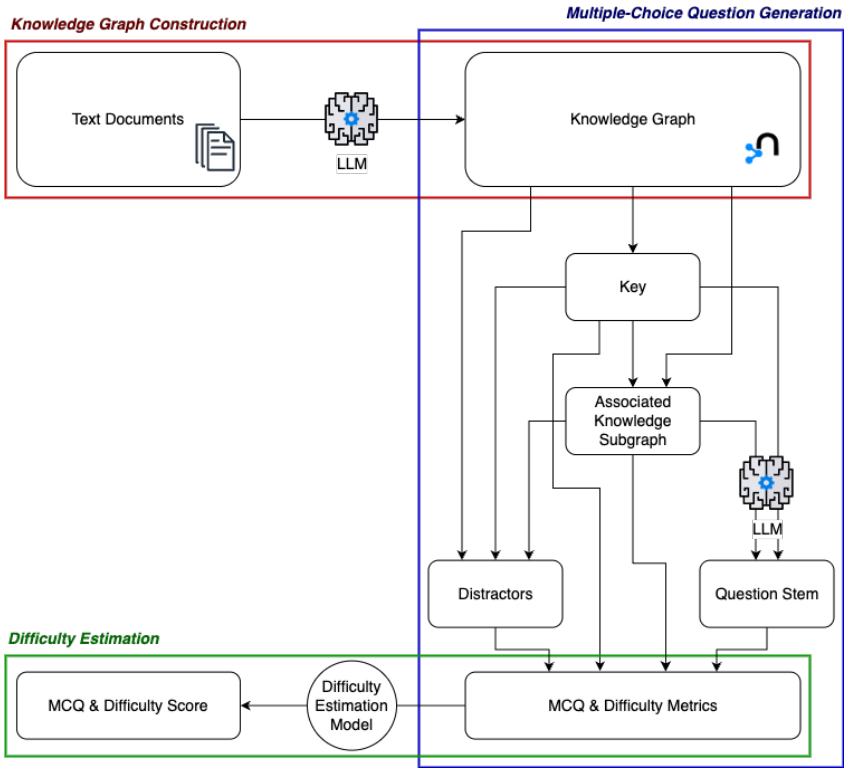

Figure 3. Overview of the proposed framework.

3.1. Knowledge Graph Construction

Given a set of input documents, we construct a KG to serve as the foundation for MCQ generation. To extract structured information from the given documents, we apply a large language model (LLM) to identify salient factual statements and convert them into subject–predicate–object triples. These triples capture entity–relation–entity structures that represent core knowledge units within the text. The procedure is summarized in Algorithm 1, which outlines how each document is processed into a graph document and subsequently integrated into a unified KG stored in a Neo4j (https://neo4j.com, accessed on 3 April 2026) database.

Algorithm 1 Overview of the Knowledge Graph Construction Process from Input Documents Using an LLM and Integration into a Neo4j Graph Database.

Input: Set of documents $\mathcal{D} = \{doc_1, doc_2, \ldots, doc_n\}$
for each $doc_i \in \mathcal{D}$ **do**
    $graphDocument_i \leftarrow$ `LLMGraphTransformer`(doc_i) ▷ Extract nodes with types and relationships using an LLM, specifically with LangChain's LLMGraphTransformer
end for
$\mathcal{G} \leftarrow \{graphDocument_1, \ldots, graphDocument_n\}$ ▷ Aggregate all extracted graph documents
`constructAndSaveKG`$(\mathcal{G})$ ▷ Parse and persist the graph in Neo4j

We begin by constructing the knowledge graph (KG) from a curated collection of textual documents. In this work, we use Wikipedia's top-100 most-viewed articles (https://en.wikipedia.org/wiki/Wikipedia:Popular_pages#Top-100_list, accessed 3 April 2026), which we automatically retrieved using a custom scraping script. The raw content is processed to extract factual knowledge using GPT-4o (https://platform.openai.com/docs/models/gpt-4o, accessed 3 April 2026), a large language model provided by OpenAI.

To transform unstructured text into structured graph representations, we employ `LLMGraphTransformer` from LangChain (https://www.langchain.com, accessed 3 April 2026).This tool ingests a list of documents, and using the underlying LLM, it extracts subject–predicate–object triples to represent key factual relationships. These triples are subsequently stored in a Neo4j graph database. Thus, the KG captures the semantic entities, their types, and the relations between them, which are later leveraged to generate MCQs.

The KG construction pipeline is designed to be schema-free, accommodating a wide variety of relations and entity types extracted from natural language. To facilitate downstream tasks, we also compute a node centrality metric, specifically degree centrality, and node embeddings for each entity in the graph using FastRP by Chen et al. [27], which later serve as useful components for some of the difficulty signals. The entire construction process is automated, resulting in a rich and semantically meaningful representation of the source content. Although the schema-free extraction is useful for open-domain text and avoids requiring a manually designed ontology in advance, it may introduce risks such as inconsistent relation names, duplicate entities, overly broad or vague entity types, incorrect relation directions, and unsupported or hallucinated facts. The proposed pipeline partially mitigates these through node-name and label-name correction and node merging. However, the above-mentioned errors can still propagate downstream by producing misleading stems, allowing invalid distractors, or distorting graph-based signals such as degree centrality, graph distance, and node-embedding similarity.

To quantify this, we performed an author-side screening for the 299 KG edges used in MCQ generation. Both endpoints were found in the used Wikipedia text for 278 of these edges (93.0%). The first screen was intentionally strict: it required local source text, and therefore should be read as a deterministic, first-pass support test rather than as the final KG correctness rate; 161 of 299 edges (53.8%) were directly supported, while 138 of 299 edges (46.2%) required second-stage review. Since the deterministic screen can miss indirectly expressed relations, alias mismatches, and relations supported outside the top local context, we added `gpt-5-mini`-assisted verification for the 138 second-stage cases, using only retrieved passages from the Wikipedia text used. This LLM-assisted audit labeled 97 of the 138 cases as supported (70.2%), 40 as unsupported by the retrieved evidence (29.0%), and 1 as unclear (0.7%). Combining these two screens, 258 of 299 (86.3%) KG-edges were supported. The same calculation over unique subject–predicate–object edges produced similar rates: 238 of 274 (86.9%) were supported, indicating that the result is not driven by repeated edge instances. Among the 40 unsupported `gpt-5-mini` second-stage cases, the LLM classified 22 as *entity/evidence-missing* (within the Wikipedia texts), which may reflect the LLM's parametric knowledge used while generating KG, 10 as *relation-unsupported*, 3 as *wrong-direction*, and 5 as *other* evidence/type issues.

3.2. Multiple-Choice Question Generation

Once the knowledge graph is constructed, we systematically generate MCQs by selecting high-centrality nodes as key candidates. Specifically, in this work, we choose the top-40 most central nodes in the KG based on degree centrality and attempt to generate four MCQs per node. The process that defines the generation of a single MCQ is outlined in Algorithm 2. Each MCQ is generated from either a triple (single-hop question), a triple with

additional context from an extra triple, a quintuple (double-hop question), or a quintuple with additional context from an extra triple. Formally, a triple denotes one KG relation involving the key node and another entity, together with their entity labels and relation direction. A quintuple denotes a two-hop path with three entities and two relations, again retaining entity labels and edge directions. The proposed pipeline attempts to generate four forms for each key node: (1) single-hop, (2) single-hop with a helper triple, (3) double-hop, and (4) double-hop with a helper triple. Candidate triples and two-hop paths are selected subject to repetition constraints; i.e., no duplicate entities. When multiple eligible triples or paths exist, the pipeline samples among them randomly.

Algorithm 2 Overview of the MCQ generation process: The pipeline samples an associated subgraph from the KG given the key, prompts an LLM to generate a question stem from it, retrieves proper graph-based distractors, computes difficulty signals of the generated MCQ after validation, and saves the resulting MCQ with its difficulty signals.

```
Input: kg                                   ▷ Knowledge graph constructed in Algorithm 1
Input: keyNode                              ▷ Selected node to serve as the correct answer
tripleOrQuintuple ← Sample a triple or quintuple of keyNode
useExtraTriple ← Set to True or False
if useExtraTriple then
    extraTriple ← Retrieve another triple of keyNode
    associatedSubgraph ← {tripleOrQuintuple, extraTriple}
else
    associatedSubgraph ← {tripleOrQuintuple}
end if
prompt ← constructPrompt(keyNode, associatedSubgraph)
questionStem ← LLM(prompt)
maxDepth ← 5                                ▷ Maximum BFS depth for distractor search
distractors ← generateDistractors(kg, keyNode,
                  associatedSubgraph, maxDepth)
if validate(questionStem, keyNode, distractors) then
    difficultySignals ← Compute the defined difficulty signals
    mcq ← MCQ(questionStem, keyNode, distractors)
    save(mcq, difficultySignals)
end if
```

For each selected node (key), a related triple or quintuple is selected from the KG. Optionally, an extra triple may be added to provide additional context. These structured elements are passed to a prompt-based question generation module powered by GPT-4o to create the question stem. The prompts are crafted to follow the style of well-crafted trivia questions, inspired by formats such as those used in the quiz show *Who Wants to Be a Millionaire?*. Figure 4 presents illustrative examples of each subgraph type alongside potential question stems that could be generated from them.

Distractors are selected from the KG using a breadth-first search (BFS) strategy, ensuring they belong to the same semantic type, i.e., to the same label after label normalization, as the key and are progressively distant in graph depth. The BFS is configured to collect eligible candidates by undirected graph distance from the key up to maximum depth 5, after which three distractors are sampled from the eligible set and randomly assigned to answer positions. In rare cases, the KG does not contain a sufficient number of appropriate distractors for a given key. In such instances, the MCQ generation attempt is aborted to preserve the quality of the generated MCQ. In our experiments, this limitation accounts for a total of 156 finalized MCQs instead of the originally targeted 160. Since the current experiments do not include a controlled comparison against random same-type or embedding-ranked distractor selection, BFS cannot be said to be superior to these. Instead, in our pipeline, BFS is used as a judicious, graph-grounded heuristic.

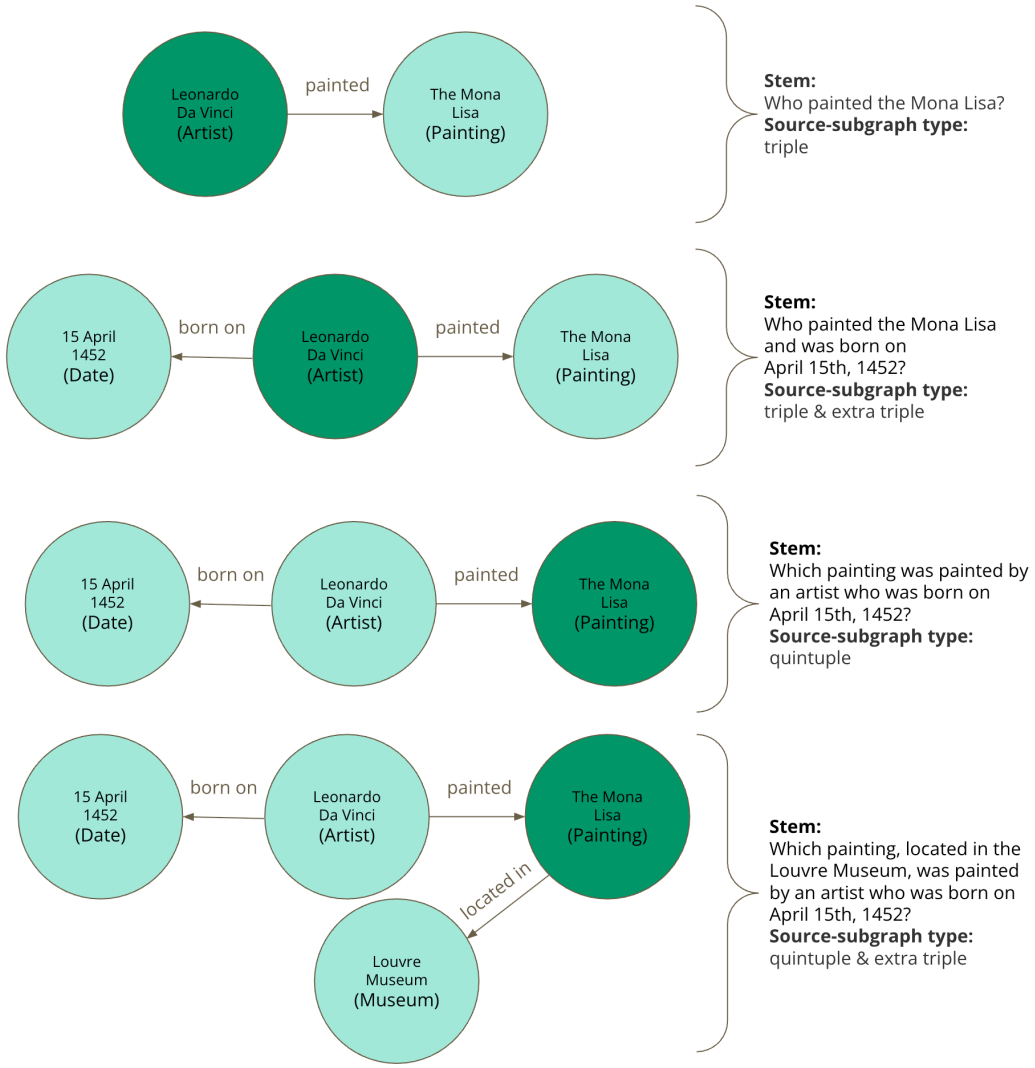

Figure 4. Illustrative examples of each subgraph type alongside potential question stems that could be generated from them. The green node represents the key in each case.

At this stage, having obtained the key, the question stem, and the distractors, the construction of the MCQ is complete. To ensure the quality and correctness of the generated MCQ, we perform a post-processing step for validation using a large language model. Each distractor option is independently evaluated by prompting the LLM with the question stem and the respective distractor. If the model identifies any distractor as a potentially correct answer, the MCQ is marked as invalid. This automated validation process serves as a safeguard against ambiguities or inconsistencies in the question formulation, ensuring that each MCQ has a single, unambiguous correct answer. Only those questions that pass this validation step are retained for subsequent analysis.

In cases where an MCQ fails the validation step, a retry mechanism is triggered to attempt regeneration. The process revisits the same key entity and initiates a new question generation sequence, potentially selecting a different supporting triple, quintuple, or extra triple, and generating a new set of distractors. This mechanism is repeated up to a predefined number of retries to ensure robustness and increase the likelihood of producing a valid MCQ. If all attempts fail, the MCQ is discarded from the final dataset to maintain overall quality. Table 3 summarizes the details for reproducibility with dataset references.

Table 3. Reproducibility parameters for the experiments.

Component	Setting Recovered from Code
Source documents	Wikipedia top-100 popular pages scraped into `wikipedia_popular_pages_dataset.json`.
KG extraction	LLMGraphTransformer with gpt-4o; temperature set to 0 in `lc_kgc.py`.
Graph database	Neo4j; graph is saved as `popular_wikipedia_dataset_gpt4o-graph-docs-v1.pkl`.
Post-processing	Node-name correction, label-name correction, and node merging using the scripts listed in the released code.
Node embeddings	Neo4j GDS FastRP, undirected graph projection, embedding dimension 256, random seed 42.
Centrality	Neo4j GDS degree centrality as node properties.
MCQ-generation model	gpt-4o through the local OpenAI wrapper in `utils/llm.py`.
MCQ-generation temperature	The wrapper stores a default temperature of 1, but the OpenAI chat-completion call does not pass the temperature argument; OpenAI API defaults therefore apply unless rerun.
Validation prompt	Each distractor is independently classified from the stem and candidate answer using the labels `CORRECT` or `FALSE`.
Retry budget	3 recursive retries per generation form after LLM validation failure.
Distractor search	Same-label candidate filter, fact/path exclusion, undirected BFS from the key, maximum depth 5, then random sampling of three eligible distractors.
Regression seed	Fixed seed 2.

As a final MCQ quality screening, we used five LLMs as our AI-based participants and made them answer all 156 candidate MCQs using only the stem and four answer options. The evaluated models were gpt-4o-mini, gpt-4o, gpt-5-mini, gpt-5.4-mini, and gpt-5.5. For gpt-4o-mini and gpt-4o, we used temperature 0 and JSON output mode. For the gpt-5 family models, the API rejected temperature 0, so the default temperature was used with JSON output mode. The models achieved accuracies of 94.9%, 98.1%, 94.9%, 96.2%, and 96.8%, respectively. Six MCQs were missed by at least 2/5 models. From a universality perspective, this is an expected failure mode of KG-based MCQ generation rather than merely a model error: unless one has access to an exhaustive closed-world oracle containing all relevant facts, a generator cannot guarantee 100% item correctness when the key and distractors share the same answer type and lie in close graph-theoretic proximity. KGs are typically incomplete, and under the open-world assumption, an unobserved relation cannot safely be treated as false [28,29]. At the same time, MCQ writing guidelines deliberately favor plausible, homogeneous, semantically close distractors while requiring exactly one correct answer [30,31]; when these constraints coincide, option separability can become underdetermined without exhaustive external verification. To increase question quality and to reduce model dependency, they were removed from the experiments.

Finally, all 150 MCQs are saved with a set of metadata features referred to as difficulty signals, which are later utilized for difficulty estimation. All signals are normalized into the range $[0, 1]$ using min-max normalization to ensure consistency and comparability across different feature types. These signals serve as interpretable proxies for MCQ difficulty and are further investigated and explained in Section 3.3, where we also describe how they are integrated into a unified difficulty estimation model.

3.3. Difficulty Estimation

To estimate the difficulty of each generated MCQ, we extract and analyze a set of nine interpretable signals, each capturing a distinct aspect of the question's structure, semantics, or linguistic complexity. These signals detailed here serve both as standalone indicators and as input features to a regression model for predicting difficulty scores.

Reasoning: This binary signal distinguishes between questions that are derived from a single triple (single-hop reasoning) and those constructed from a quintuple (double-hop reasoning). Double-hop questions are generally more cognitively demanding, as they require the integration of multiple relational facts. Figure 5 illustrates this distinction.

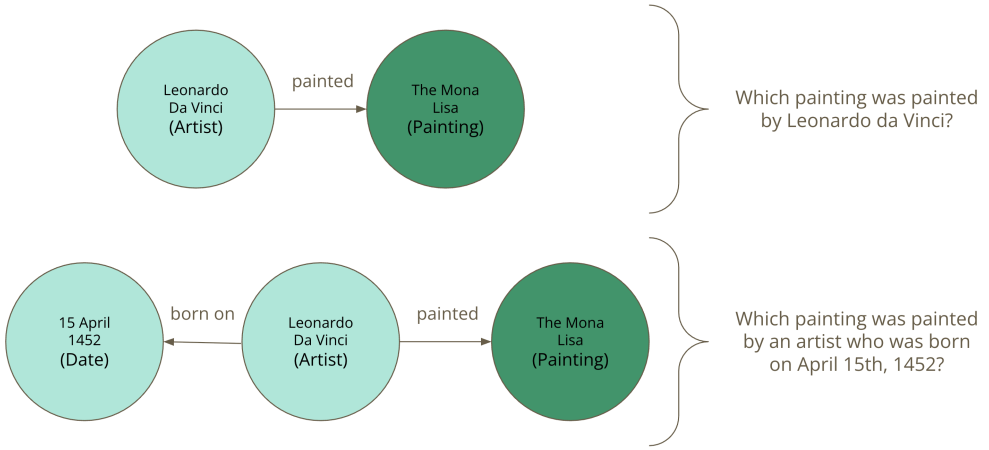

Figure 5. Illustration of single-hop and double-hop reasoning for MCQ generation. The green node represents the key.

Extra Triple: This binary signal captures whether an additional supporting triple is incorporated, providing further context. The inclusion of an extra triple may enrich the given knowledge inside the MCQ, but also potentially increase its complexity. An example is shown in Figure 6.

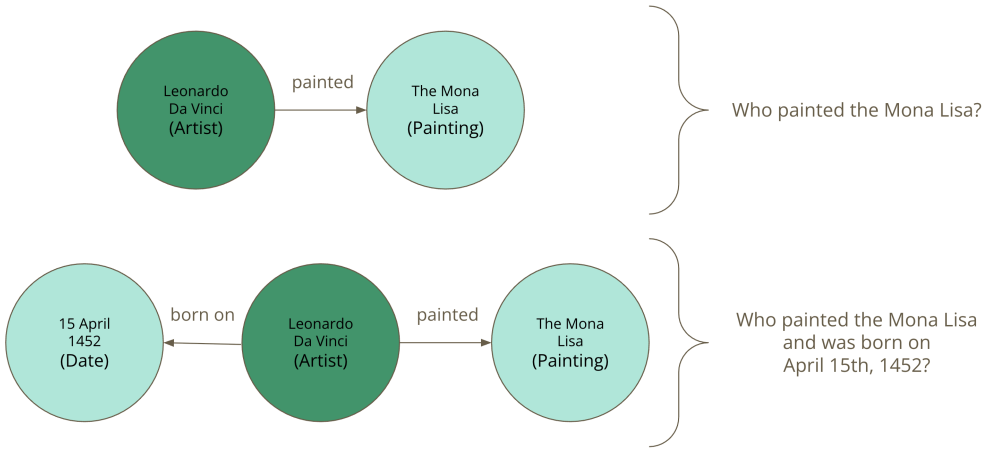

Figure 6. Illustration of MCQ generation with and without an extra triple. The green node represents the key.

Distractor Depth: This signal measures the average depth of distractors from the key entity in the knowledge graph using breadth-first traversal. The farther the distractors are from the key entity, the less semantically related they tend to be, which typically reduces the difficulty. This structural notion is visualized in Figure 7.

Node Embedding Similarity: We compute the average cosine similarity between the node embeddings of the distractors and the key, using FastRP embeddings, denoted as e_n. This is expressed in (1).

$$\frac{1}{3}\sum_{i=1}^{3}\cos(e_n(\text{node}(d_i)), e_n(\text{node}(key))) \quad (1)$$

Higher similarity implies more plausible distractors, increasing the MCQ's difficulty.

Text Embedding Similarity This signal captures the semantic similarity between each distractor and the question stem in the embedding space. To contextualize the score, it is normalized by the similarity between the key and the stem. It is computed as

$$\frac{\frac{1}{3}\sum_{i=1}^{3}\cos(e_t(\text{text}(d_i)), e_t(\text{stem}))}{\cos(e_t(\text{text}(key)), e_t(\text{stem}))} \tag{2}$$

where e_t is the embedding function, specifically OpenAI's text-embedding-3-large (https://platform.openai.com/docs/models/text-embedding-3-large, accessed on 3 April 2026) model.

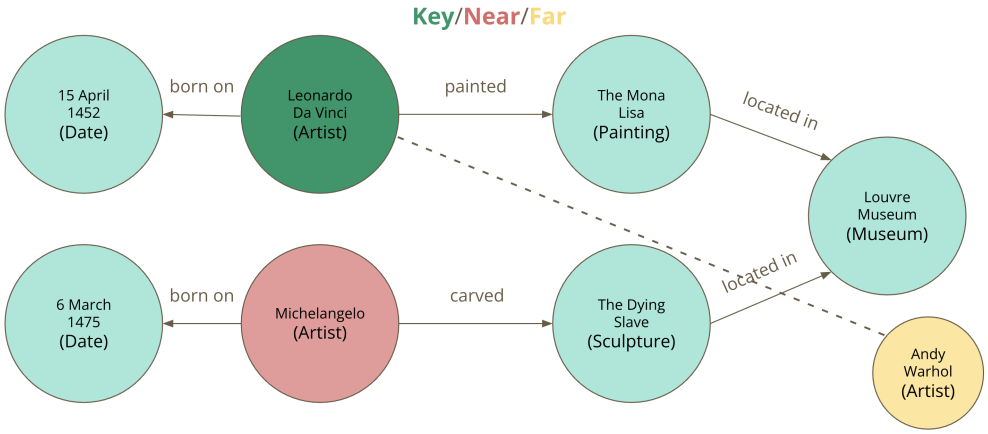

Figure 7. Visualization of candidate distractor depths. The green node represents the key. The dashed line represents a path.

Degree Centrality: The degree centrality of a node v is defined as the number of incoming and outgoing edges it has. The formal definition is given in (3).

$$\frac{1}{|V_s|}\sum_{v\in V_s}\deg(v). \tag{3}$$

The signal takes the average degree centrality of all nodes involved in the associated knowledge subgraph. Higher centrality often implies more well-known concepts.

Readability: We compute the Flesch Reading Ease score, denoted as $\mathtt{F}$, for the question stem, following the formulation proposed by Flesch [32]. This score is used as the readability signal to quantify linguistic complexity. The computation is given below:

$$\mathtt{F} = 206.835 - 1.015 \cdot \left(\frac{\#\text{ words}}{\#\text{ sentences}}\right) - 84.6 \cdot \left(\frac{\#\text{ syllables}}{\#\text{ words}}\right). \tag{4}$$

Above Largest Gap Count: We compute the cosine similarity between the question stem and each of the four answer options, including the key and the three distractors, and sort the resulting similarity scores in descending order: $\text{sim}_1 \geq \text{sim}_2 \geq \text{sim}_3 \geq \text{sim}_4$. We then calculate the differences between adjacent similarity values and identify the position of the largest gap, drawing inspiration from the Maximum Gap Thresholding technique proposed by Soykök and Güvenir [33]. The revised definition is a two-step computation rather than a direct count returned by an arg max. First, we identify the largest adjacent semantic gap:

$$g^* = \arg\max_{i\in\{1,2,3\}}(\text{sim}_i - \text{sim}_{i+1}). \tag{5}$$

The signal is then obtained via a min–max normalization of g^* to the $[0, 1]$ interval. It captures how many answer options are semantically grouped close to the stem before a separation point, potentially influencing the difficulty of identifying the correct answer.

LLM Extra Fact: This binary signal indicates whether the generated question stem includes factual content not directly inferable from the associated knowledge subgraph. Conceptually, this signal is partly a validity/quality-control indicator and partly a possible difficulty

predictor. If a stem introduces unsupported facts, participants may need outside knowledge or may face ambiguity unrelated to the selected KG subgraph. Such additions may also be unintended hallucinations by the LLM and typically increase the difficulty or ambiguity of the question. Thus, a negative correlation with incorrect-answer rate means that detected unsupported extra content is associated with greater empirical difficulty. This signal is automatically generated for each MCQ by prompting an LLM to compare the question stem and the associated knowledge subgraph. As Table 4 shows, the effect is modest, supporting the interpretation of this signal primarily as a diagnostic quality-control feature rather than as a dominant predictor.

Table 4. Distribution of and empirical difficulty by the LLM Extra Fact signal.

Encoded Value	MCQs	Incorrect-Rate Mean	Incorrect-Rate SD
0: unsupported extra fact detected	80	0.340	0.189
1: no unsupported extra fact detected	70	0.304	0.178
Correlation with incorrect rate		−0.097	

All signals are normalized to the $[0, 1]$ interval using min-max normalization to ensure scale compatibility. Together, they serve as interpretable and complementary indicators of MCQ difficulty. Moving on, we explore how these signals are leveraged to train a supervised model for estimating difficulty scores. Table 5 summarizes the proposed signals.

Table 5. Overview of MCQ-generation signals and their descriptions.

Signal	Description
Reasoning	A binary indicator denoting whether the MCQ was derived from a triple or a quintuple.
Extra Triple	A binary indicator representing the use of an extra triple for additional context.
Distractor Depth	The average graph distance between the key and each distractor.
Node Embedding Similarity	The average cosine similarity between the key and distractors based on their graph node embeddings.
Text Embedding Similarity	The average cosine similarity between the distractors and the question stem, divided by the cosine similarity between the key and the stem.
Degree Centrality	The average degree centrality of the entities involved in the question stem, reflecting their graph-level prominence.
Readability	The Flesch Reading Ease score of the question stem, indicating its linguistic complexity.
Above Largest Gap Count	A normalized version of the ordinal indicator capturing the number of distractor options that precede the largest semantic gap when cosine similarities between the question stem and each distractor are sorted in descending order.
LLM Extra Fact	A binary diagnostic flag encoded as 0 when unsupported extra factual content is detected in the question stem and 1 when none is detected.

Difficulty Estimation Model

After computing the defined difficulty signals for each validated MCQ, we train a supervised regression model to estimate the MCQ difficulty. The target variable is the empirical difficulty score, approximated by the observed incorrect response rate collected from users. This score lies within the $[0, 1]$ interval, where higher values correspond to

more difficult questions. The proposed difficulty signals are model-agnostic and can be straightforwardly used as input features in any regression framework.

The resulting regression model provides a continuous-valued difficulty score for each MCQ, which we refer to as its *estimated difficulty*. In addition to enabling quantitative difficulty estimation, this approach allows us to interpret the relative importance of each difficulty signal through feature importance analysis. Since the dataset is small, and the generated items are clustered by key node, our evaluation does not rely only on a single split. We performed k-fold cross-validation and report the mean performance for each model as well as the standard deviations.

4. Dataset

To the best of our knowledge, there exists no available dataset that jointly provides (i) the KG or subgraph used to construct a multiple-choice question (MCQ), (ii) the MCQ itself, and (iii) a corresponding difficulty label. Given that our difficulty estimation framework relies on both the structural and semantic characteristics of the source knowledge graph as well as the inherent properties of the MCQ itself, we constructed a dedicated dataset for this task. We began by collecting textual data from Wikipedia's top-100 most popular articles, which we automatically retrieved using a custom scraping script. These documents were then processed using LangChain's `LLMGraphTransformer` with OpenAI's GPT-4o model to construct a knowledge graph. The resulting KG captures semantic relationships and entity types, and serves as the foundation for MCQ generation.

To generate MCQs, we selected the top-40 most central nodes in the graph using degree centrality and attempted to generate four MCQs per node using our methodology. We presented all the generated 156 MCQs to a student participant pool, and for each MCQ, we received approximately 38 responses on average. Six of the generated MCQs, which were in the human study, are removed from the analysis in this paper after an LLM-based evaluation as described above. This yielded a total of 150 MCQs. The raw human-study table contains 5938 submitted responses from 216 participants across all 156 generated MCQs. Most participants completed 30 questions, and each MCQ received a mean of 38.06 responses, with a minimum of 24 and a maximum of 53. After excluding the six weakly answerable MCQs, the final 150-item set used in the analyses contains 5684 responses, with a mean of 37.89 responses per MCQ.

For each MCQ, we computed the *incorrect answer rate*, the proportion of participants who answered the question incorrectly, and used it as the empirical difficulty score. Figure 8 shows the distribution of incorrect-answer rates across the final 150-item dataset. The mean incorrect response rate was 0.323 with a standard deviation of 0.184, indicating a moderate overall difficulty level after the answerability-based exclusions. The distribution is slightly skewed toward easier questions, likely due to the intentional selection of high-centrality nodes in the KG, to ensure familiarity and contextual relevance for participants. The present dataset should therefore be interpreted as a common-knowledge benchmark. Additionally, participants were asked to rate how much they liked each question on a standardized scale. These ratings were averaged to yield a quality metric for each MCQ. The mean liking score across all MCQs was 66%, indicating a generally favorable reception. Also, the negative association between average liking and empirical difficulty was statistically significant: $r = -0.49$, $p = 1.96 \times 10^{-10}$, and 95% CI is $[-0.603, -0.358]$, indicating that participants tended to prefer easier questions.

To understand how the proposed difficulty signals relate to the (empirical) question difficulty, Table 6 presents the Pearson correlation between each normalized signal and the empirical incorrect-answer rate. The strongest association is observed for Text Embedding Similarity ($r = 0.471$), suggesting that MCQs tend to become more difficult when the

distractors are semantically close to the question stem relative to the key. This finding supports the intuition that semantically plausible distractors increase participant confusion. Above Largest Gap Count also shows a moderate positive correlation ($r = 0.299$), indicating that when multiple answer options remain semantically close to the stem before a clear separation point, identifying the correct answer becomes more challenging.

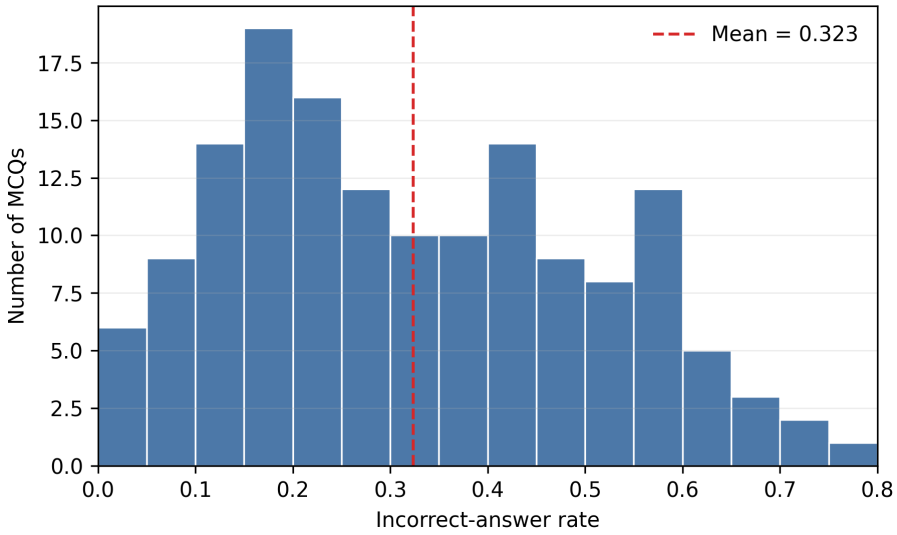

Figure 8. Histogram of the incorrect-answer rates of the final 150-item analyzed MCQ set.

Table 6. Pearson correlation between each difficulty signal and empirical incorrect-answer rate.

Signal	Correlation (r)
Text Embedding Similarity	0.471
Above Largest Gap Count	0.299
Distractor Depth	0.142
Degree Centrality	−0.135
Reasoning	0.116
LLM Extra Fact	−0.097
Node Embedding Similarity	0.085
Extra Triple	0.063
Readability	−0.045

In addition to Table 6, Table 7 shows that although they do not represent exactly the same thing, several difficulty signals are correlated with each other, which is important for interpreting the subsequent predictive models. The strongest pairwise correlation is between Distractor Depth and Node Embedding Similarity ($r = 0.634$), indicating that graph distance and graph-embedding proximity capture related structural information from the KG. Similarly, Text Embedding Similarity and Above Largest Gap Count are strongly correlated ($r = 0.592$), which is expected because both signals are derived from semantic similarities between the question stem and the answer options.

Overall, the dataset, whose statistics are summarized in Table 8, offers a unique resource where each data point consists of the source knowledge subgraph, the generated MCQ, and an observed incorrect-answer rate as the difficulty score. It enables modeling the difficulty using both KG-based and MCQ-based signal sources, effectively addressing our initial requirement.

Table 7. Largest absolute Pearson correlations among difficulty signals.

Signal 1	Signal 2	Correlation (r)
Distractor Depth	Node Embedding Similarity	0.634
Text Embedding Similarity	Above Largest Gap Count	0.592
Extra Triple	Degree Centrality	−0.512
Extra Triple	LLM Extra Fact	−0.349
Distractor Depth	Text Embedding Similarity	0.342
Reasoning	Text Embedding Similarity	0.339
Reasoning	Readability	−0.250
Degree Centrality	Text Embedding Similarity	−0.213

Table 8. Dataset statistics for the final analyzed MCQ set.

Statistic	Value
Final MCQs	150
Unique key nodes	39
MCQs per key node	3–4
Single-hop/single-hop + helper	39/39
Double-hop/double-hop + helper	38/34
Incorrect-rate mean ± SD	0.323 ± 0.184
Incorrect-rate range	0.000–0.750

5. Experimental Results and Analysis

This section presents the experimental evaluation of our proposed difficulty estimation framework. All experiments were conducted on a MacBook Pro equipped with an Apple M3 chip, integrated GPU, and 16 GB of unified memory. The system was running macOS with ARM64 architecture. We evaluate several standard regression models, specifically Linear Regression, Random Forest, and Gradient Boosting from the scikit-learn (https://scikit-learn.org/, accessed on 7 May 2026) library, as well as XGBoost from the XGBoost (https://xgboost.readthedocs.io/, accessed on 7 May 2026) library (all with default parameters). We report evaluation metrics including root mean squared error (RMSE), mean absolute error (MAE), coefficient of determination (R^2 score), and Spearman's Rank Correlation, as defined below.

5.1. Evaluation Metrics

- Root Mean Squared Error (RMSE): Measures the square root of the average squared differences between predicted and actual values. RMSE penalizes larger errors more than smaller ones.
- Mean Absolute Error (MAE): Represents the average absolute difference between predicted and actual values, offering a more interpretable error measure.
- Coefficient of Determination (R^2): Indicates the proportion of variance in the actual difficulty scores explained by the predicted scores.
- Spearman's Rank Correlation (ρ): Measures the strength and direction of the monotonic relationship between predicted and actual difficulty scores. This metric is particularly useful for evaluating how well the predicted rankings of MCQs align with the ground-truth rankings, thus reflecting the model's ability to preserve the relative ordering in terms of MCQ difficulty.

These metrics collectively provide a comprehensive view of model accuracy, robustness, and consistency in ranking, which are important for assessing prediction quality.

5.2. Model Selection and Results

Given the limited size of our dataset, we restrict our analysis to the classical regression models stated above. Table 9 summarizes the 5-fold cross-validation performance of these models. Furthermore, to visualize the prediction behavior of each model, we use out-of-fold predictions from the same 5-fold cross-validation procedure. For each MCQ, the plotted prediction is produced only by the fold model for which that MCQ was held out during training. The plots are given in Figure 9. The metrics shown are the out-of-fold metrics over the final 150-item analyzed set and are therefore comparable in scale to the cross-validation results in Table 9. Based on these results, the Random Forest regressor performs slightly better than the other alternatives.

Table 9. Five-fold cross-validation results for the final analyzed MCQ set. Values are fold-averaged mean ± standard deviation, i.e., each metric is first computed separately on each validation fold and then averaged.

Model	RMSE	MAE	R^2	Spearman's ρ
5-fold CV				
Linear Regression	0.175 ± 0.023	0.146 ± 0.017	0.019 ± 0.230	$28.3 \pm 13.7\%$
Random Forest	0.171 ± 0.025	0.141 ± 0.022	0.046 ± 0.290	$37.0 \pm 20.0\%$
Gradient Boosting	0.186 ± 0.034	0.154 ± 0.029	-0.149 ± 0.425	$32.0 \pm 24.9\%$
XGBoost	0.174 ± 0.030	0.142 ± 0.029	0.018 ± 0.324	$35.0 \pm 22.9\%$

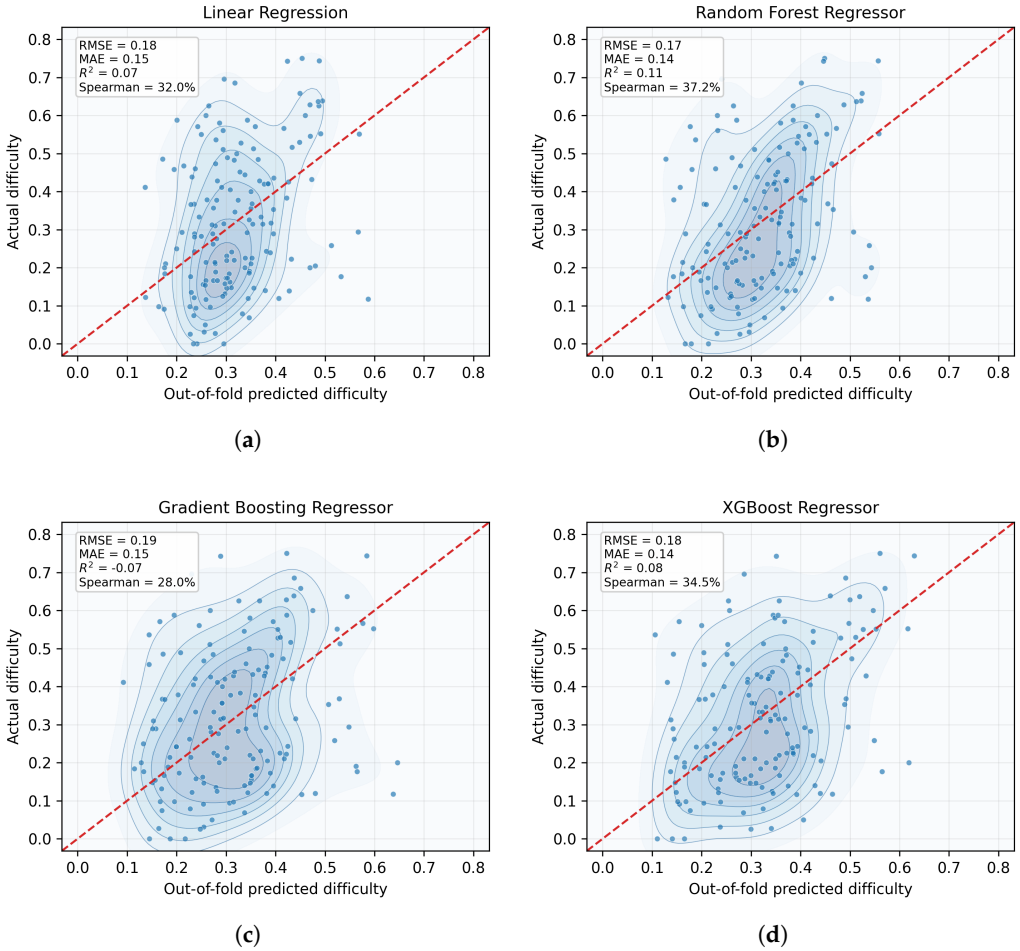

Figure 9. Predicted vs. actual difficulty for each regression model using strict out-of-fold predictions from 5-fold cross-validation. Each MCQ is predicted only by the fold model for which it was held out during training. (**a**) Linear Regression. (**b**) Random Forest Regressor. (**c**) Gradient Boosting Regressor. (**d**) XGBoost Regressor. The dashed, red lines correspond to $x = y$.

5.3. Feature Importance Analysis

Since 5-fold cross-validation identifies Random Forest as the strongest, we base the feature-importance and ablation analyses with RF as the base. Figure 10 reports both *impurity-based* (left) feature importances from a model trained on the full final 150-item analyzed set and cross-validated *permutation importances* (right). The impurity-based values on the left quantify the average reduction in prediction impurity attributed to each feature across the decision trees used in the RF ensemble.

Permutation importances use the same 5-fold cross-validation used for model comparison, i.e., Table 9. In each fold, we permute each feature on the held-out fold (the corresponding 20%) 30 times, and importance was measured as the average increase in RMSE after permutation. This analysis reveals how much predictive accuracy is lost when a feature/signal is disrupted after training. Both views of feature importance identify the Text Embedding Similarity as the dominant signal. The impurity-based importances also assign weight to Readability, Node Embedding Similarity, Degree Centrality, and Distractor Depth, whereas the permutation analysis seems to be more conservative.

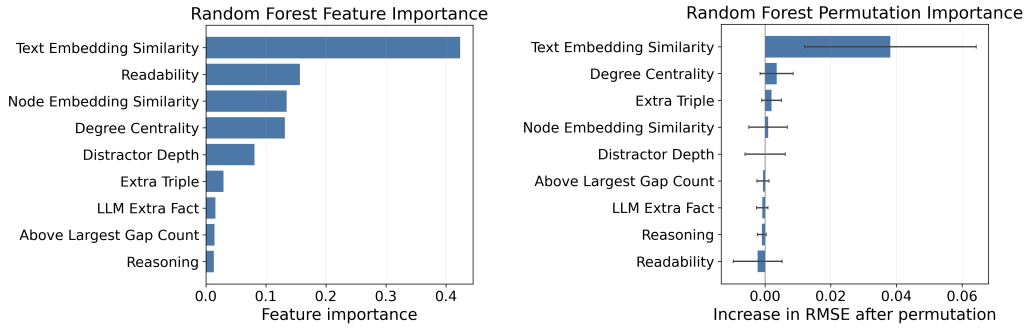

Figure 10. Feature-importance analyses (based on RF) on the final 150-item analyzed set. (**a**) shows impurity-based feature importances from a model trained on the full analyzed set. (**b**) shows cross-validated permutation importance, measured as the mean increase in RMSE after feature permutation across 5 folds and 30 repeats per fold, with error bars denoting the standard deviation.

When interpreted together with the marginal and pairwise correlations in Tables 6 and 7, Text Embedding Similarity is the most stable difficulty signal: it has the strongest marginal association with incorrect-answer rate ($r = 0.471$) and is also the dominant feature in both the impurity-based and permutation-based RF importance analyses. This implies that semantic plausibility between the stem, the key, and the distractors is the main driver of MCQ difficulty. Above Largest Gap Count has the second-largest marginal correlation with difficulty ($r = 0.299$), but its incremental model importance is small, which is due to its strong pairwise correlation with Text Embedding Similarity ($r = 0.592$). Degree Centrality has a negative marginal correlation with difficulty ($r = -0.135$) as expected. However, the analysis reveals that it is not the strongest difficulty-increasing signal (which can be a concern since the questions are generated for the keys chosen due to their high centrality).

Readability is the least stable signal. The impurity-based RF importance assigns it noticeable weight, but its marginal difficulty correlation is near zero. Furthermore, its permutation importance is negligible. Therefore, it needs to be interpreted cautiously: it may help a tree model partition the small dataset, but the present evidence does not support a strong individual claim about its predictive role. As the future section reveals, it acts as a noise in the one-out ablation study we performed for completeness.

5.4. Ablation Study

To perform an exhaustive ablation study, we evaluated all candidate feature subsets of size $f \in \{9, 8, 7, 6, 5, 4\}$ under the same 5-fold cross-validation used. Table 10 reports the strongest RF configuration for each f value. Based on these results, the 5-feature RF model, with Extra Triple, Degree Centrality, Distractor Depth, Node Embedding Similarity, and Text Embedding Similarity, performs the best. Hence, most of the proposed features, including both graph- and text-based ones, were useful in the prediction/ranking performance.

Table 10. RF feature-set search results under shuffled 5-fold cross-validation on the final 150-item analyzed set. For each feature count, each row gives the lowest-RMSE RF subset of that size. Values are fold-averaged mean ± standard deviation, computed by evaluating each metric separately on each validation fold and then averaging across folds. The row in bold is the best configuration.

f	Included Features/Signals	RMSE	MAE	R^2	Spearman's ρ
9	All nine signals	0.171 ± 0.025	0.141 ± 0.022	0.046 ± 0.290	37.0 ± 20.0%
8	All except Readability	0.166 ± 0.026	0.136 ± 0.023	0.113 ± 0.243	40.6 ± 15.7%
7	Ext. Trip.; Deg. Cent.; Distr. Depth; Node Emb. Sim.; Text Emb. Sim.; LLM Extra Fact; Abv. Gap Cnt.	0.166 ± 0.024	0.136 ± 0.022	0.117 ± 0.234	42.3 ± 15.7%
6	Ext. Trip.; Deg. Cent.; Distr. Depth; Node Emb. Sim.; Text Emb. Sim.; Abv. Gap Cnt.	0.164 ± 0.025	0.134 ± 0.021	0.132 ± 0.233	42.7 ± 15.5%
5	Ext. Trip.; Deg. Cent.; Distr. Depth; Node Emb. Sim.; Text Emb. Sim.	**0.163 ± 0.025**	**0.133 ± 0.022**	**0.139 ± 0.245**	**43.5 ± 17.0%**
4	Ext. Trip.; Node Emb. Sim.; Text Emb. Sim.; LLM Extra Fact	0.170 ± 0.025	0.141 ± 0.023	0.071 ± 0.246	40.6 ± 13.6%

The remaining four features, Readability (with the lowest correlation to the incorrect-answer rate, as Table 6 shows), LLM Extra Fact, Reasoning, and Above Largest Gap Count, were not selected. Table 11 shows the performance indicators with a classical 1-out ablation study: Removing Readability improves all metrics relative to the all-signal baseline, reducing RMSE from 0.171 to 0.166, increasing R^2 from 0.046 to 0.113, and increasing Spearman's ρ from 37.0% to 40.6%. Furthermore, as expected, among the retained features, Text Embedding Similarity is indispensable. Similarly, leaving out Degree Centrality, Extra Triple, or Node Embedding Similarity has a negative contribution to RMSE. The only feature remaining is Distractor Depth. Overall, the feature importance and ablation analysis jointly suggest that the best RF model benefits from a compact combination of semantic similarity, graph structure, and contextual subgraph information.

The feature-set search indicates that several signals are redundant in the current small dataset. The strongest RF configuration used five signals and improved the average RMSE from 0.171 to 0.163. However, because these subset choices are selected after evaluating many alternatives on the same 150-item dataset, the results should be interpreted as exploratory evidence of redundancy rather than as final feature selection proof.

Figure 11 compares strict out-of-fold predictions from the all-signal and the five-signal RF models. Both models use the same Random Forest settings and the same shuffled 5-fold cross-validation split. The five-signal model uses Extra Triple, Degree Centrality, Distractor Depth, Node Embedding Similarity, and Text Embedding Similarity. In pooled out-of-fold evaluation, the all-signal model achieved $RMSE = 0.173$, $MAE = 0.141$, $R^2 = 0.108$, and Spearman's $\rho = 37.2\%$, while the five-signal model achieved $RMSE = 0.165$, $MAE = 0.133$, $R^2 = 0.189$, and Spearman's $\rho = 42.3\%$.

Table 11. Single-feature ablation results for the Random Forest Regressor under 5-fold cross-validation on the final 150-item analyzed set. The first row gives the all-signal baseline; each remaining row removes one signal. Values are fold-averaged mean ± standard deviation across validation folds. The row in bold is the best configuration.

Removed Signal	RMSE	MAE	R^2	Spearman's ρ
None (all signals)	0.171 ± 0.025	0.141 ± 0.022	0.046 ± 0.290	37.0 ± 20.0%
Reasoning	0.171 ± 0.026	0.141 ± 0.022	0.048 ± 0.285	37.9 ± 18.9%
Extra Triple	0.173 ± 0.026	0.143 ± 0.023	0.019 ± 0.309	35.2 ± 22.1%
Degree Centrality	0.176 ± 0.025	0.146 ± 0.021	−0.008 ± 0.305	35.0 ± 19.0%
Distractor Depth	0.171 ± 0.025	0.139 ± 0.022	0.062 ± 0.254	39.2 ± 17.5%
Node Embedding Similarity	0.172 ± 0.029	0.142 ± 0.024	0.024 ± 0.354	42.6 ± 22.5%
Text Embedding Similarity	0.197 ± 0.019	0.165 ± 0.018	−0.247 ± 0.268	4.8 ± 17.9%
LLM Extra Fact	0.171 ± 0.026	0.141 ± 0.022	0.048 ± 0.288	36.7 ± 21.6%
Readability	**0.166 ± 0.026**	**0.136 ± 0.023**	**0.113 ± 0.243**	**40.6 ± 15.7%**
Above Largest Gap Count	0.171 ± 0.026	0.141 ± 0.023	0.055 ± 0.283	37.4 ± 21.2%

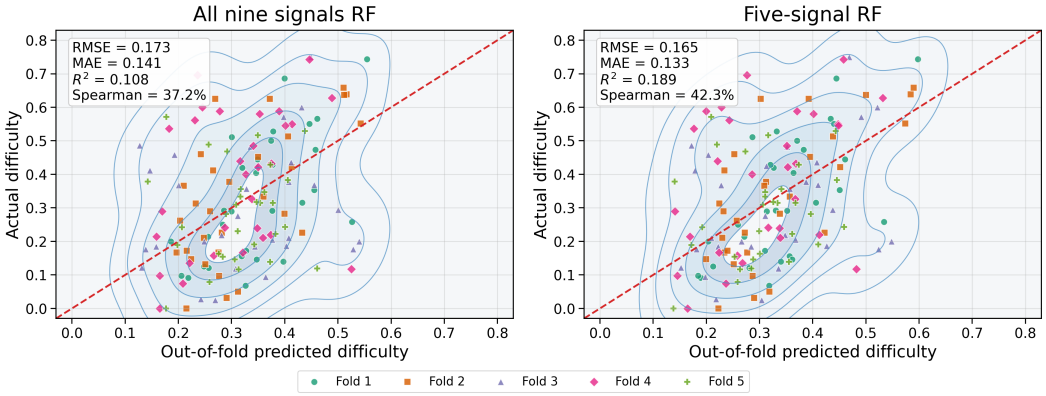

Figure 11. Strict out-of-fold predicted vs. actual difficulty for the all-signal and the selected five-signal RF models under shuffled 5-fold CV. The markers and colors distinguish the held-out fold ID. The metric boxes report pooled out-of-fold metrics, computed once over all 150 MCQs. The dashed, red lines correspond to $x = y$.

5.5. Group-Aware Validation by Key Entity

Since each key was used to generate multiple (3/4) MCQs, a random split can place MCQs with the same key in both training and validation. This creates a possible leakage concern. However, the difficulty model does not receive the key name, identifier, source article, or option text as direct input. Most of the features are relative (i.e., the difference/similarity) or aggregate quantities, such as key–distractor graph distance, key–distractor node-embedding similarity, stem–option text-embedding similarity, and subgraph form. There exist numerical features such as Degree Centrality, which is computed from the properties of *all* the entities involved in the stem, not the key itself. Therefore, the model cannot directly memorize that a specific entity, such as a particular actor or country, is easy or difficult. The remaining risk is distributional rather than a direct leakage: MCQs generated around the same key can share similar local KG neighborhoods and distractor-construction behavior.

To assess this risk, we added a group-aware 5-fold validation in which all MCQs sharing the same correct-answer key entity are assigned to the same held-out fold. Figure 12 shows the corresponding strict out-of-fold predictions for the all-signal and selected five-signal RF models using the group assignment. The results remain in the same general range as the traditional 5-fold analysis: the all-signal RF achieved pooled out-of-fold $RMSE = 0.170$, $MAE = 0.139$, $R^2 = 0.138$, and Spearman's $\rho = 40.9\%$, while

the five-signal RF achieved $RMSE = 0.167$, $MAE = 0.135$, $R^2 = 0.167$, and Spearman's $\rho = 42.7\%$. This robustness check suggests that the shuffled 5-fold results are not solely driven by direct key-identity memorization.

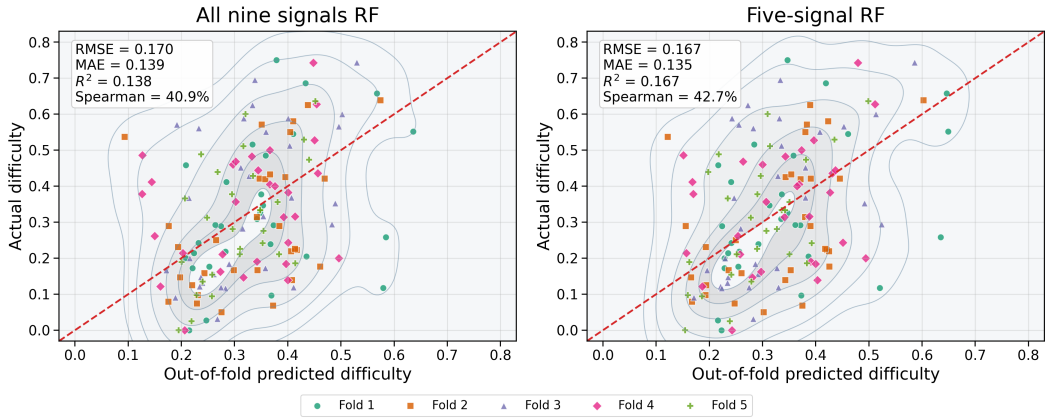

Figure 12. Group-aware 5-fold out-of-fold predicted vs. actual difficulty for the all-signal and selected five-signal RF models. All MCQs sharing the same correct-answer key entity are assigned to the same held-out fold. Markers and colors indicate the held-out fold. The metric boxes report pooled out-of-fold metrics, computed once over all 150 MCQs. The dashed, red lines correspond to $x = y$.

6. Discussion

The 5-fold CV results in Table 9 identify Random Forest as the best model by average RMSE and MAE on the final 150-item analyzed set, although the margin over XGBRegressor is small. This indicates that the proposed signals contain usable difficulty-related information, yet the predictive effect is sensitive to the small sample size. The exhaustive feature-set search provides a more systematic view of signal redundancy than the earlier one-signal-at-a-time ablation. After removing the six weakly answerable MCQs, the all-signal RF model achieved $RMSE = 0.171 \pm 0.025$, $MAE = 0.141 \pm 0.022$, $R^2 = 0.046 \pm 0.290$, and Spearman's $\rho = 37.0 \pm 20.0\%$ under shuffled 5-fold CV. The strongest five-signal subset, consisting of Extra Triple, Degree Centrality, Distractor Depth, Node Embedding Similarity, and Text Embedding Similarity, reduced the average error to $RMSE = 0.163 \pm 0.025$ and $MAE = 0.133 \pm 0.022$. In pooled out-of-fold evaluation with the same split, this five-signal model achieved $RMSE = 0.165$, $MAE = 0.133$, $R^2 = 0.189$, and Spearman's $\rho = 42.3\%$. Thus, the compact RF model improves both absolute error metrics and rank ordering in this dataset. As a robustness check, the group-aware validation in Section 5.5 produced similar performance for the key-group split, suggesting that the shuffled 5-fold pattern is not solely explained by direct leakage.

The feature importance analyses further clarify that Text Embedding Similarity is consistently the strongest signal, and assign nontrivial importance to Readability, Node Embedding Similarity, Degree Centrality, and Distractor Depth. However, the permutation analysis shows that most non-text-similarity signals have unstable effects. Even so, several signals appear useful in combination. LLM Extra Fact exhibits relatively minimal predictive influence, supporting its interpretation primarily as a diagnostic quality-control signal rather than as a dominant difficulty predictor. Taken together, these findings support the viability of the proposed signals for interpretable exploratory difficulty estimation. This being said, they do not yet support a stable predictive accuracy across new domains, new source corpora, or unseen key-node distributions, which require larger datasets.

The quality-control audits also affect how the results should be interpreted. The final experiments exclude six MCQs that at least two independent LLMs failed to answer correctly from the provided stem and options. This exclusion improves the methodological

focus of the difficulty-modeling analysis, because the target becomes the difficulty of answerable MCQs rather than the difficulty induced by erroneous MCQs. At the same time, this finding is consistent with the literature showing that the automatic validation of distractor quality is difficult under the open-world assumption—yet LLMs trained with huge datasets may be leveraged rather than the expensive, manual expert audit (or to filter and make the human intervention less costly).

This study indicates that interpretable difficulty estimation for automatic, KG-based MCQ generation is feasible in the analyzed proof-of-concept setting through a combination of semantic, structural, and linguistic features. However, several limitations remain: the practical deployment in real-world educational and assessment contexts may require larger datasets, domain-specific evaluation, and expert review. The KG is also automatically extracted by an LLM; our strict and LLM-assisted audits show that most of the KG edges are supported by the saved source text, but 40 of 299 audited edges in the final 150-MCQ analyzed set (13.4%) remained unsupported. Second, the LLM is used at multiple stages, including KG construction, stem generation, and validation, which can introduce correlated LLM-dependent errors. The main observed failure modes involve unsupported KG relations. Overall, this study treats the pipeline as novel, analyzable, and promising but not yet deployment-ready. However, it is also expected to be more robust with better, faster, open- and closed-weight LLMs appearing every year.

7. Conclusions and Future Work

This study introduces a novel, proof-of-concept framework for generating multiple-choice knowledge questions with interpretable difficulty estimation by integrating knowledge graphs (KGs), which provide structured representations of factual knowledge, with large language models (LLMs), which offer advanced natural language understanding and generation capabilities. Starting from unstructured textual sources, we construct a KG that captures factual relationships, generate MCQs from selected subgraphs, and compute a suite of interpretable signals that reflect both structural and linguistic aspects of each MCQ. These signals are then used as input features to train regression models for predicting empirical difficulty scores, demonstrating reasonably strong alignment with human evaluations.

Our experimental results confirm the viability of this promising approach, showing that difficulty can be effectively estimated using features derived from both the underlying KG and the MCQs themselves. Under shuffled 5-fold cross-validation, RF achieved the best average RMSE and MAE among the evaluated models, with only a small margin over XGBoost. An exhaustive RF feature-set search identified a compact five-signal subset—Extra Triple, Degree Centrality, Distractor Depth, Node Embedding Similarity, and Text Embedding Similarity—as the strongest configuration. In pooled out-of-fold evaluation under the same shuffled 5-fold protocol, this model achieved $RMSE = 0.165$, $MAE = 0.133$, $R^2 = 0.189$, and Spearman's $\rho = 42.3\%$. Overall, the results show that the proposed signals, especially semantic-similarity features, contain useful difficulty-related information.

The current experimental results support the feasibility and analyzability of the pipeline, while deployment-scale validity, domain transfer, and expert-certified educational quality remain possible and important directions for future work. We are planning to scale the proposed approach with diverse data and more comprehensive KGs. Furthermore, we will test whether the same difficulty signals generalize for the difficulty estimation on external public exam datasets under different response models.

Author Contributions: Conceptualization, H.A.G. and K.K.; methodology, M.C.Ş.; software, M.C.Ş.; validation, M.C.Ş. and H.A.G.; data curation, M.C.Ş.; writing—original draft preparation, M.C.Ş., H.A.G. and K.K.; writing—review and editing, M.C.Ş., H.A.G. and K.K.; writing—revision preparation, H.A.G. and K.K.; visualization, M.C.Ş., H.A.G. and K.K.; supervision, H.A.G.; project administration, H.A.G. and K.K.; funding acquisition, H.A.G. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the European Union under the Horizon Europe Research and Innovation Action (RIA) project titled Customized Games and Routes for Cultural Heritage and Arts (Grant Agreement No. 101094428) (https://cordis.europa.eu/project/id/101094428, accessed on 3 April 2026). The authors gratefully acknowledge the funding provided by the European Commission, which enabled the development of this study. The views and opinions expressed are those of the authors and do not necessarily reflect those of the European Union or the granting authority.

Institutional Review Board Statement: This study, related to our dataset collection experiment, has been approved by the Bilkent University IRB with approval number 662 on 1 March 2025.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study. The human evaluation was conducted with university students from Bilkent University (aged 18–22). Participation was voluntary, and no incentives were provided. To avoid any perceived pressure or conflict of interest, students of the authors were not eligible to participate in the study. No demographic information was collected. Participants responded to the generated MCQs and rated how much they liked each question on a standardized scale.

Data Availability Statement: The data and the required scripts and prompts are available at: https://doi.org/10.5281/zenodo.20067933 (accessed on 3 April 2026).

Conflicts of Interest: The authors have no relevant financial or non-financial interests to disclose.

Abbreviations

The following abbreviations are used in this manuscript:

MCQ	Multiple-choice question
KG	Knowledge graph
LLM	Large language model
RNN	Recurrent neural network
GNN	Graph neural network
GAT	Graph attention network
SPARQL	SPARQL Protocol and RDF Query Language
NER	Named entity recognition
BFS	Breadth-first search
MoE	Mixture of experts
BiLSTM	Bidirectional long short-term memory
RTRL	Relation-aware transformer with reinforcement learning
KGNN-ADP	Knowledge graph neural network-based adaptive difficulty prediction
AutoDE	Automatic difficulty estimation
FastRP	Fast random projection
RMSE	Root mean squared error
MAE	Mean absolute error
R^2	Coefficient of determination
RF	Random forest
CV	Cross-validation

References

1. Reddy, S.; Raghu, D.; Khapra, M.M.; Joshi, S. Generating Natural Language Question-Answer Pairs from a Knowledge Graph Using a RNN Based Question Generation Model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*; Lapata, M., Blunsom, P., Koller, A., Eds.; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 376–385.
2. Elsahar, H.; Gravier, C.; Laforest, F. Zero-Shot Question Generation from Knowledge Graphs for Unseen Predicates and Entity Types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*; Walker, M., Ji, H., Stent, A., Eds.; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 218–228. [CrossRef]
3. Chen, Y.; Wu, L.; Zaki, M.J. Toward Subgraph-Guided Knowledge Graph Question Generation with Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 12706–12717. [CrossRef] [PubMed]
4. Li, Z.; Cao, Z.; Li, P.; Zhong, Y.; Li, S. Multi-Hop Question Generation with Knowledge Graph-Enhanced Language Model. *Appl. Sci.* **2023**, *13*, 5765. [CrossRef]
5. Alsubait, T.; Parsia, B.; Sattler, U. Ontology-Based Multiple Choice Question Generation. *KI Künstliche Intell.* **2016**, *30*, 183–188. [CrossRef]
6. Seyler, D.; Yahya, M.; Berberich, K. Knowledge Questions from Knowledge Graphs. In Proceedings of the ICTIR '17: ACM SIGIR International Conference on Theory of Information Retrieval, New York, NY, USA, 1–4 October 2017; pp. 11–18. [CrossRef]
7. Kumar, V.; Hua, Y.; Ramakrishnan, G.; Qi, G.; Gao, L.; Li, Y.F. Difficulty-Controllable Multi-hop Question Generation from Knowledge Graphs. In *Proceedings of the The Semantic Web—ISWC 2019*; Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F., Eds.; Springer: Cham, Switzerland, 2019; pp. 382–398.
8. Kusuma, S.F.; Siahaan, D.O.; Fatichah, C. Automatic question generation with various difficulty levels based on knowledge ontology using a query template. *Knowl. Based Syst.* **2022**, *249*, 108906. [CrossRef]
9. Bi, S.; Liu, J.; Miao, Z.; Min, Q. Difficulty-controllable question generation over knowledge graphs: A counterfactual reasoning approach. *Inf. Process. Manag.* **2024**, *61*, 103721. [CrossRef]
10. AlKhuzaey, S.; Grasso, F.; Payne, T.R.; Tamma, V. Text-based Question Difficulty Prediction: A Systematic Review of Automatic Approaches. *Int. J. Artif. Intell. Educ.* **2023**, *34*, 862–914. [CrossRef]
11. Zhu, J.; Liu, D.; Chen, S. Multiple-choice question generation and difficulty calculations based on semantic similarity. *Neural Comput. Appl.* **2025**, *37*, 13295–13307. [CrossRef]
12. Wei, L.; Hao, G.S. Knowledge Graph Based Absolute Difficulty Prediction of Multiple Choice Question. In Proceedings of the 2024 9th International Conference on Electronic Technology and Information Science (ICETIS), Hangzhou, China, 17–19 May 2024; pp. 720–727. [CrossRef]
13. Leo, J.; Kurdi, G.; Matentzoglu, N.; Parsia, B.; Sattler, U.; Forge, S.; Donato, G.; Dowling, W. Ontology-Based Generation of Medical, Multi-term MCQs. *Int. J. Artif. Intell. Educ.* **2019**, *29*, 145–188. [CrossRef]
14. Kumar, A.P.; Nayak, A.; K, M.S.; Chaitanya.; Ghosh, K. A Novel Framework for the Generation of Multiple Choice Question Stems Using Semantic and Machine-Learning Techniques. *Int. J. Artif. Intell. Educ.* **2023**, *34*, 332–375. [CrossRef]
15. Huang, W.; Zhou, G.; Lapata, M.; Vougiouklis, P.; Montella, S.; Pan, J.Z. Prompting large language models with knowledge graphs for question answering involving long-tail facts. *Knowl. Based Syst.* **2025**, *324*, 113648. [CrossRef]
16. Jiang, B.; Wang, Y.; Luo, Y.; He, D.; Cheng, P.; Gao, L. Reasoning on Efficient Knowledge Paths: Knowledge Graph Guides Large Language Model for Domain Question Answering. In Proceedings of the 2024 IEEE International Conference on Knowledge Graph (ICKG), Abu Dhabi, United Arab Emirates, 11–12 December 2024; pp. 142–149. [CrossRef]
17. Vachev, K.; Hardalov, M.; Karadzhov, G.; Georgiev, G.; Koychev, I.; Nakov, P. Leaf: Multiple-Choice Question Generation. In *Proceedings of the Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, 10–14 April 2022, Proceedings, Part II*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 321–328. [CrossRef]
18. Zeng, H.; Wei, B.; Liu, J. RTRL: Relation-aware Transformer with Reinforcement Learning for Deep Question Generation. *Knowl. Based Syst.* **2024**, *300*, 112120. [CrossRef]
19. Johnson, B.G.; Van Campenhout, R.; Jerome, B.; Castro, M.F.; Bistolfi, R.; Dittel, J.S. Automatic Question Generation for Spanish Textbooks: Evaluating Spanish Questions Generated with the Parallel Construction Method. *Int. J. Artif. Intell. Educ.* **2025**, *35*, 1044–1063. [CrossRef]
20. Gao, Y.; Bing, L.; Chen, W.; Lyu, M.; King, I. Difficulty Controllable Generation of Reading Comprehension Questions. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, Macao, China, 10–16 August 2019*; International Joint Conferences on Artificial Intelligence Organization: Los Angeles, CA, USA; pp. 4968–4974. [CrossRef]
21. Cheng, Y.; Li, S.; Liu, B.; Zhao, R.; Li, S.; Lin, C.; Zheng, Y. Guiding the Growth: Difficulty-Controllable Question Generation through Step-by-Step Rewriting. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021*; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Kerrville, TX, USA: 2021; pp. 5968–5978. [CrossRef]

22. Grévisse, C.; Pavlou, M.A.S.; Schneider, J.G. Docimological Quality Analysis of LLM-Generated Multiple Choice Questions in Computer Science and Medicine. *SN Comput. Sci.* **2024**, *5*, 636. [CrossRef]
23. Artsi, Y.; Sorin, V.; Konen, E.; Glicksberg, B.S.; Nadkarni, G.; Klang, E. Large Language Models for Generating Medical Examinations: Systematic Review. *BMC Med. Educ.* **2024**, *24*, 354. [CrossRef] [PubMed]
24. Feng, W.; Lee, J.; McNichols, H.; Scarlatos, A.; Smith, D.; Woodhead, S.; Ornelas, N.; Lan, A. Exploring Automated Distractor Generation for Math Multiple-choice Questions via Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, 16–21 June 2024; pp. 3067–3082. [CrossRef]
25. Zotos, L.; van Rijn, H.; Nissim, M. Are You Doubtful? Oh, It Might Be Difficult Then! Exploring the Use of Model Uncertainty for Question Difficulty Estimation. In Proceedings of the 18th International Conference on Educational Data Mining, Palermo, Italy, 20–23 July 2025.
26. Awalurahman, H.W.; Budi, I. Automatic Distractor Generation in Multiple-Choice Questions: A Systematic Literature Review. *PeerJ Comput. Sci.* **2024**, *10*, e2441. [CrossRef] [PubMed]
27. Chen, H.; Sultan, S.F.; Tian, Y.; Chen, M.; Skiena, S. Fast and Accurate Network Embeddings via Very Sparse Random Projection. In Proceedings of the CIKM '19: 28th ACM International Conference on Information and Knowledge Management, New York, NY, USA, 3–7 November 2019; pp. 399–408. [CrossRef]
28. Shi, B.; Weninger, T. Open-World Knowledge Graph Completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [CrossRef]
29. Yang, H.; Lin, Z.; Zhang, M. Rethinking Knowledge Graph Evaluation Under the Open-World Assumption. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022. [CrossRef]
30. Haladyna, T.M.; Downing, S.M.; Rodriguez, M.C. A Review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment. *Appl. Meas. Educ.* **2002**, *15*, 309–333. [CrossRef]
31. Mitkov, R.; Ha, L.A.; Varga, A.; Rello, L. Semantic Similarity of Distractors in Multiple-Choice Tests: Extrinsic Evaluation. In Proceedings of the Workshop on GEometrical Models of Natural Language Semantics, Athens, Greece, 30–31 March 2009; pp. 49–56.
32. Flesch, R. A new readability yardstick. *J. Appl. Psychol.* **1948**, *32*, 221–233. [CrossRef] [PubMed]
33. Soykök, I.T.; Güvenir, H.A. Multi-label multi-modal classification of movie scenes. *Knowl. Based Syst.* **2025**, *318*, 113459. [CrossRef]