applied sciences

Article

TwinStar: A Novel Design for Enhanced Test Question Generation Using Dual-LLM Engine

Qingfeng Zhuge *, Han Wang and Xuyang Chen

School of Computer Science and Technology, East China Normal University, Shanghai 200063, China
* Correspondence: qfzhuge@cs.ecnu.edu.cn

Abstract: In light of the remarkable success of large language models (LLMs) in natural language understanding and generation, a trend of applying LLMs to professional domains with specialized requirements stimulates interest across various fields. It is desirable to further understand the level of intelligence that can be achieved by LLMs in solving domain-specific problems, as well as the resources that need to be invested accordingly. This paper studies the problem of generating high-quality test questions with specified knowledge points and target cognitive levels in AI-assisted teaching and learning. Our study shows that LLMs, even those as immense as GPT-4 or Bard, can hardly fulfill the design objectives, lacking clear focus on cognitive levels pertaining to specific knowledge points. In this paper, we explore the opportunity of enhancing the capability of LLMs through system design, instead of training models with substantial domain-specific data, consuming mass computing and memory resources. We propose a novel design scheme that orchestrates a dual-LLM engine, consisting of a question generation model and a cognitive-level evaluation model, built with fine-tuned, lightweight baseline models and prompting technology to generate high-quality test questions. The experimental results show that the proposed design framework, TwinStar, outperforms the state-of-the-art LLMs for effective test question generation in terms of cognitive-level adherence and knowledge relevance. TwinStar implemented with ChatGLM2-6B improves the cognitive-level adherence by almost 50% compared to Bard and 21% compared to GPT-4.0. The overall improvement in the quality of test questions generated by TwinStar reaches 12.0% compared to Bard and 2% compared with GPT-4.0 while our TwinStar implementation consumes only negligible memory space compared with that of GPT-4.0. An implementation of TwinStar using LLaMA2-13B shows a similar trend of improvement.

Keywords: automated test generation; large language model; fine-tuning; prompting

Academic Editor: Pedro Couto

Received: 24 December 2024
Revised: 16 February 2025
Accepted: 21 February 2025
Published: 12 March 2025

Citation: Zhuge, Q.; Wang, H.; Chen, X. TwinStar: A Novel Design for Enhanced Test Question Generation Using Dual-LLM Engine. *Appl. Sci.* **2025**, *15*, 3055. https://doi.org/10.3390/app15063055

1. Introduction

Large language models (LLMs), such as GPT-4 [1,2], Gemini [3], Claude [4], LLaMA [5], and ChatGLM [6,7], have brought transformative changes across various application domains, including machine translation [8], text summarization [9], sentiment analysis [10], medical advice consultation [11,12], and legal document analysis [13,14], with powerful capabilities in natural language understanding and generation [15]. A trend of applying LLMs to professional domains with specialized requirements that stimulates research interest across various fields also casts a profound impact on the field of education. LLM-based AI technology not only provides reading or writing assistance [16,17] but also aims to be involved in the process of teaching and learning [16,18,19]. Specifically, the technique of generating high-quality test questions exhibits sophisticated intelligence, emphasizing that the purpose of a test in teaching processes is to evaluate the development curve of cognitive levels for students in certain learning stages. Our study, as well as previous research [19,20], shows that LLMs, even those as immense as GPT-4 and Gemini, can hardly fulfill the design objectives of test question generation, lacking clear focus on cognitive levels pertaining to specific knowledge points and also exposing the limitations of intelligence displayed by LLMs. An interesting question is whether an LLM is able to generate high-quality test questions without a substantial question bank. It is also desirable to further understand the resources that need to be invested accordingly in order to achieve a certain level of intelligence in solving domain-specific problems. A meaningful exploration of the possibilities of upgrading the intelligence demonstrated by LLMs will become an important step toward highly effective AI-assisted teaching and learning.

This paper studies the problem of generating high-quality test questions with specified knowledge points and target cognitive levels. As oppose to the well-studied classical test design problem, also known as the optimal test design on a question bank with multiple objectives, the test question generation problem studied in this paper focuses on the connection between content and cognitive levels that can be rendered by LLM-based AI systems for educational purposes. Based on the widely recognized Bloom's taxonomy [21], there are six major cognitive levels in the learning process. These levels encompass remembering, understanding, applying, analyzing, evaluating, and creating, progressing from lower-order to higher-order thinking. It is critical to use questions targeting various cognitive levels to assess the development of students' understanding of specific knowledge points in effective teaching; however, designing high-quality test questions that meet these requirements is usually challenging and time-consuming.

This work aims to investigate methods for enhancing LLMs' capability to generate cognitive-level-aware test questions. We explore this opportunity through system design, instead of training models with substantial domain-specific data, consuming mass computing and memory resources. We propose a framework for enhanced test question generation utilizing a dual-LLM AI engine, called TwinStar. The framework orchestrates two LLMs: a question generation model and a cognitive-level evaluation model. The question generation model generates test questions based on specified cognitive level and knowledge points with the assistance of multiple rounds of prompts. The cognitive-level evaluation model evaluates the generated test question by cognitive level and provides feedbacks interatively to the question generation model to try to realign with cognitive-level requirements. Our implementation of TwinStar, using two fine-tuned, lightweight baseline models (ChatGLM2-6B) and prompting technology, generates test questions targeting specific knowledge points in university-level calculus at designated cognitive levels without additional question bank or expert evaluation. The experimental results show that the proposed design framework, TwinStar, outperforms the state-of-the-art LLMs, including GPT-3.5, GPT-4, Claude, and Bard, for effective test question generation in terms of cognitive-level adherence and knowledge relevance. The main contributions of this paper are as follows:

- We propose a design framework for a dual-LLM AI engine (TwinStar) to enhance the capabilities of large language models in generating effective test questions with targeted knowledge points and cognitive levels. The question generation model and the cognitive-level evaluation model cooperate to achieve significant improvements in the overall quality of the generated test questions.
- By employing lightweight LLMs in a dual-LLM engine, we explore various fine-tuning and prompting techniques to build both the question generation model and the cognitive-level evaluation model. We show the progress in performance improvement achieved by multiple techniques toward the final design of TwinStar. We also find that

the evaluation model, tuned with a small set of test questions from various subjects, can effectively calibrate the cognitive level of a calculus question.

- The experimental results show that a dual-LLM engine built with two lightweight LLMs outperforms state-of-the-art LLMs, e.g., GPT-3.5, GPT-4, Claude, and Bard, in accuracy, with negligible memory consumption. The results indicate the potential of TwinStar architecture to achieve outstanding accuracy on resource-constrained systems.

This paper is organized as follows: Section 2 introduces the background and related work. Section 3 presents the design of TwinStar. Section 4 evaluates the experimental results. Finally, Section 5 concludes the paper.

2. Background and Related Work

To thoroughly discuss our challenges and design rationale, this section provides a comprehensive overview of techniques for question selection.

2.1. An Overview on LLMs

Despite the remarkable capabilities demonstrated by LLMs in natual language processing (NLP) tasks, the weakness of LLMs is also well known. First of all, it is well known that the size of most LLMSs tends to expand significantly in exchange for learning a generic representation of knowledge through a massive training process. Generally speaking, this consumes an immense amount of computing and memory resources so quickly that it is not affordable to most people and companies. Huge resource consumption obviously becomes a barrier when LLMs are employed to solve domain-specific problems, usually in a resource-constrained environment. ChatGPT-3.5, a full-fledged model, for example, has 175B parameters. The size of GPT-4 is not disclosed. An open-source baseline model, LLaMA2-13B, has 13 billion parameters, consuming about 26 GB memory space. Considering a resource-constrained environment for solving domain-specific problems, lightweight LLMs are desirable. The baseline model used in this paper, ChatGLM2-6B, only has 6B parameters, consuming about 12 GB of memory space. We consider that a large portion of the knowledge network is not relevant in solving domain-specific problems. However, a lightweight model, such as ChatGLM2, apparently struggles to fulfill this requirement by rendering a network of domain knowledge. In this paper, we show an opportunity for enhancing the capability of LLMs through system design, instead of training models with substantial domain-specific data, consuming mass computing and memory resources.

The second well-known weakness of LLMs is that the accuracy of the solution is not guaranteed because knowledge is acquired through reading comprehension. This becomes a challenge when specialized requirements need to be complied with for domain-specific problems. To enhance question generation efficacy, we employ prompting tuning (P-tuning) and multiround prompting to optimize both the question generation model and the evaluation model, creating an integrated dual-model AI engine to maintain cognitive-level adherence and knowledge relevance for the generated test questions. This study shows that the techniques adopted in this paper cast a significant impact on lightweight models. The improvement achieved in lightweight models outweighs that in large and sophisticated models.

2.2. Question Generation Approaches

Prior research on question generation has typically taken one of the following two approaches: (1) rule-based methods [22–24] and (2) neural network-based methods [25–27]. Some studies have achieved basic levels of automation in question generation using templates. For instance, Williams et al. [28] generated mathematical word problems based on ontological predicates. Ahmed et al. [29] proposed an automatic template extraction method, while Andersen et al. [23] and Singh et al. [22] developed semi-automatic template extraction methods. However, these methods heavily depend on designers' expertise in crafting templates, often resulting in rigid and repetitive question formats.

In recent years, advancements in sequence-to-sequence models, such as Recurrent Neural Networks (RNNs) and Transformers, have inspired extensive research into neural network-based question generation. For example, Li et al. [26] employed a Long Short-Term Memory (LSTM)-based model to produce elementary mathematical word problems, combining character, word, and part-of-speech tag embeddings. Similarly, Zhou et al. [25] utilized a Transformer-based architecture, where the encoder extracted topic and mathematical formula features and the decoder generated mathematical word problems. Pre-trained models, such as Bidirectional Encoder Representations from Transformers (BERT) and the Generative Pre-trained Transformer (GPT) series, have demonstrated robust performance across a range of tasks. Studies suggest that questions generated by the GPT series often outperform those from BERT-based models [30].

However, current techniques for question generation predominantly focus on mathematical word problems and reading comprehension, limiting the range of cognitive levels they address. Specifically, mathematical word problems often assess students' application skills within Bloom's taxonomy. In practical educational assessments, questions must evaluate multiple cognitive levels for specific knowledge points. Yet, existing research lacks a clear focus on generating multilevel cognitive questions tied to specific knowledge points. Consequently, prior studies offer no direct solutions to the challenges addressed in this paper.

Recent advancements in large-scale language models have created new opportunities for educational question generation. Although researchers have explored the use of ChatGPT-like models to label educational materials or assist teachers in creating them, their effectiveness remains largely unvalidated. Elkins et al. [31] examined the effectiveness of large language models in generating educational questions and found that the generated questions were generally high-quality and suitable for classroom use, though their alignment with Bloom's cognitive levels was relatively low. Other studies have attempted to generate domain-specific questions by training pre-trained models from scratch, followed by fine-tuning [20]. However, these efforts primarily evaluate question quality based on grammatical correctness without considering multilevel cognitive assessments. Moreover, these approaches often require substantial domain-specific data, which are scarce in real-world settings. Enhancing a language model's question generation capabilities for specialized domains with limited data remains an open challenge.

3. Design

The quality of test questions generally relies on domain knowledge, such as a question bank covering a substantial network of knowledge points, and a set of criteria profiling the challenge level, such as Bloom's cognitive levels. Leveraging large language models (LLMs) to generate test questions that align with educational objectives actually poses significant challenges.

First of all, the size of LLMs has a direct impact on the relevance of knowledge points for a certain subject deeply embedded in the neural network. For example, the baseline model used in this paper, ChatGLM2, has only 6B parameters. Compared with ChatGPT-3.5, a full-fledged model with 175B parameters, a single ChatGLM2 model apparently struggles to fulfill the requirement for rendering a network of domain knowledge. In our experiments, a single ChatGLM2 model only achieves 3% accuracy in knowledge relevance compared with ChatGPT-3.5. However, a lightweight model is actually desired for many domain-specific applications, providing a secure and relatively isolated environment on a local machine or an embedded system. The challenge is how to reveal domain knowledge accurately with lightweight models. In this paper, we are interested in how to accurately capture certain knowledge points in a test question generated by a relatively small model. Our work shows that a set of effective techniques, as well as a careful system design, is critical for achieving the design goal.

Secondly, a classical question generation problem usually needs to go through a process of multi-object optimization, considering both the accuracy of locating knowledge points and the level of challenge, for example. Multi-object optimization, however, is not inherent for LLMs. College-level calculus questions generated by ChatGPT-3.5, for example, only meet the requirement of Bloom's cognitive level with 78% accuracy, even though it shows an almost perfect accuracy in knowledge relevance. This means that a well-trained huge model, such as ChatGPT-3.5, can still hardly assist in the teaching process in terms of generating test questions that fit the learning curve and educational objectives. Many existing projects exercising question generation for tests still rely on a group of experts with experience in question design for a specific subject to review or evaluate the generated questions, imposing a considerable workload, which may not be affordable for various application scenarios.

To address these challenges, this paper proposes a design framework centered around a dual-LLM engine, as illustrated in Figure 1. It features two models, a question generation model and a cognitive-level evaluation model, as AI engines that collaboratively generate questions aligned with Bloom's targeted cognitive levels and knowledge points. The cognitive-level evaluation engine built on the same baseline model, ChatGLM2, assesses the alignment with Bloom's cognitive levels and provides iterative feedback to refine the direction of question generation. These two AI engines interact iteratively, akin to twin stars in orbit, which inspired the name 'TwinStar' for the proposed framework.

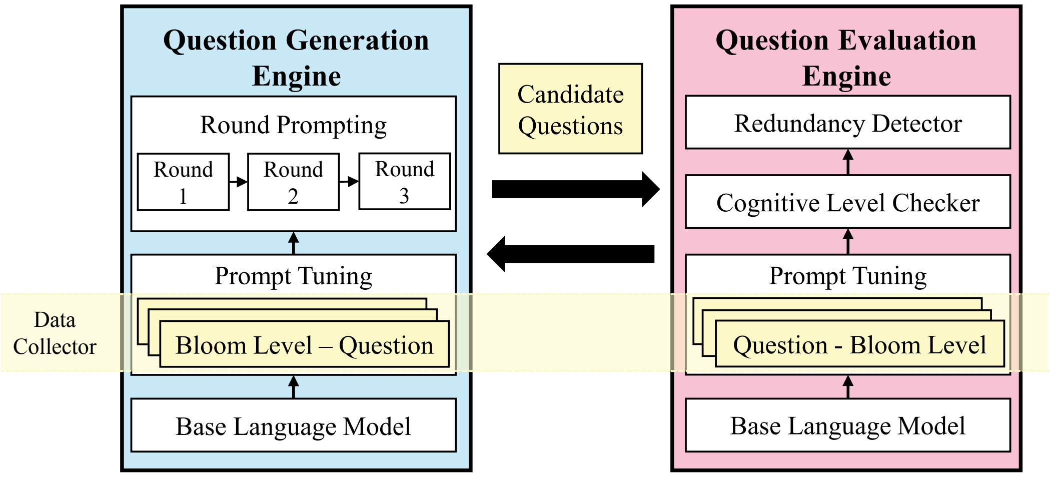


Figure 1. An overview of TwinStar.

3.1. Question Generation

3.1.1. Data Processing

Both the question generation model and the evaluation model in TwinStar underwent fine-tuning on benchmark datasets to enhance the alignment of Bloom's cognitive levels and generate questions in order to meet the design goal for accuracy. Nevertheless, for some specific subjects, such as university-level calculus, publicly available datasets that are annotated with Bloom's cognitive levels remain exceedingly scarce. Most available datasets mapped to Bloom's cognitive levels [32,33] span across diverse subjects, e.g., literature, medicine, and computer science.

In this study, we posit that the connections between Bloom's cognitive levels and the formulation of test questions are consistent across various subjects. Thus, we start with a general domain dataset for fine-tuning when no domain-specific questions are available.

Figure 2 illustrates the workflow of data collection and processing. An input instruction for the question generation model specifies the targeted Bloom's cognitive level, which instructs the model to generate a question for the specified cognitive level. An example of the constructed data is provided as follows:

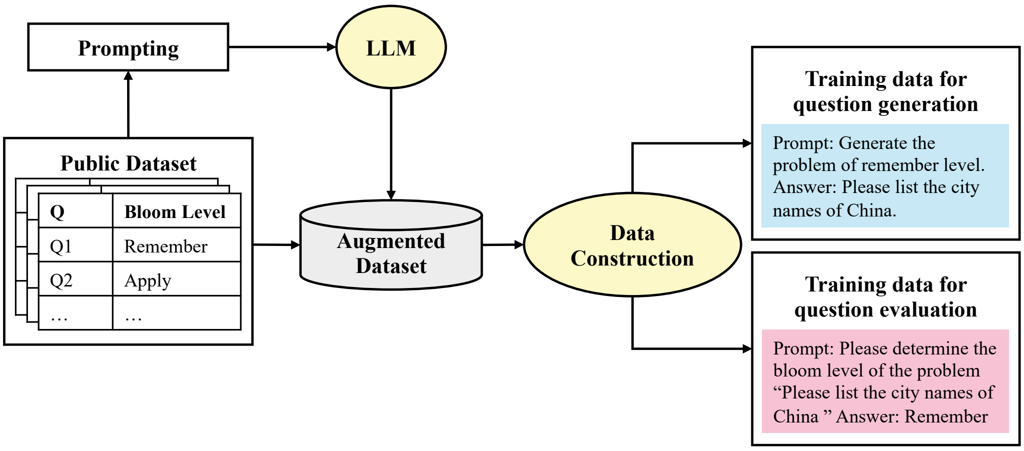

Prompting
LLM
Public Dataset
Q
Bloom Level
Q1
Remember
Q2
Apply
…
…
Augmented Dataset
Data Construction
Training data for question generation
Prompt: Generate the problem of remember level. Answer: Please list the city names of China.
Training data for question evaluation
Prompt: Please determine the bloom level of the problem "Please list the city names of China " Answer: Remember

Figure 2. The workflow of data collection and processing.

> *"Prompt": Please generate a question that belongs to the Bloom's cognitive level of "Remembering".*
>
> *"Answer": Please name 10 cities in China.*

On the other hand, the cognitive-level evaluation model is tuned to identify the corresponding Bloom's cognitive level of an input question. This indicates a reversal in the direction of data preparation. An example is shown as follows:

> *"Prompt": Please determine which Bloom's cognitive level the question "Please name 10 cities in China" belongs to.*
>
> *"Answer": "Remembering"*

Similar prompt–answer pairs are constructed in the context of college-level calculus, as shown in the following, and vice versa.

> *"Prompt": Please determine which Bloom's cognitive level the question "Use power series to show* $cos(x) + isin(x) = e^{ix}$*" belongs to.*
>
> *"Answer": "Applying"*

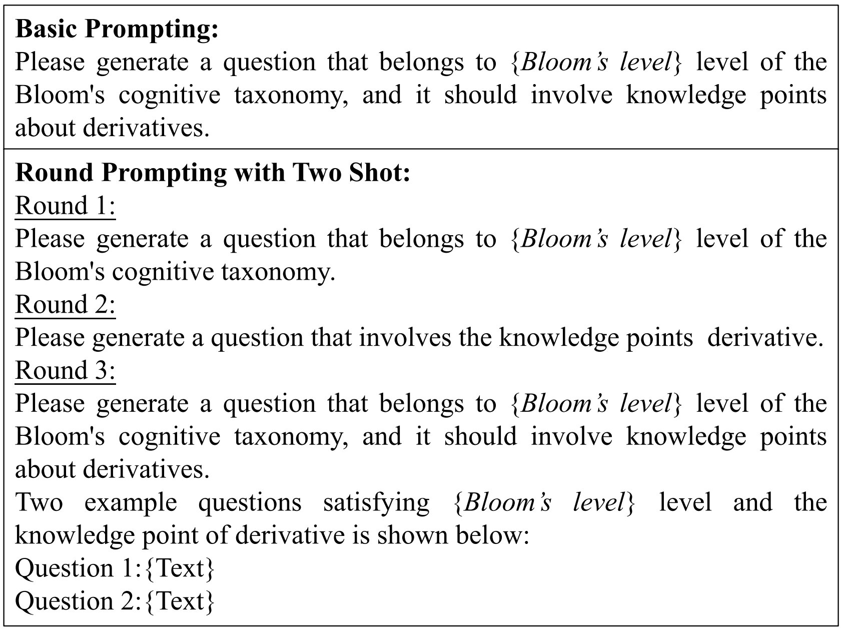
Our study shows that prompt–answer pairs constructed with Bloom's cognitive levels for various subjects can be effectively used in fine-tuning across various subject.

To enhance the effects of fine-tuning, this study employs ChatGPT-3.5 to assist in data augmentation. It generates additional questions at the analyzing, evaluating, and creating levels to make sure that the dataset for fine-tuning provides a balanced distribution across all the cognitive levels. Level-shot prompting further improves the accuracy of the data for fine-tuning. The process of data augmentation saves a lot of labor in manually expanding the training set. The augmented dataset is merged with the original dataset to fine-tune both models. This significantly reduces the overhead of manual annotation and improves the efficiency of the whole process.

3.1.2. Prompting Method

Prompting is widely used to extend the inference capability of LLMs. In this work, we develop a prompting strategy to ensure that the generated questions meet the target cognitive level, as well as specific knowledge points.

A zero-shot prompting approach is attempted at first. However, the prompts frequently fail to meet the design objectives for both the cognitive level and knowledge points at the same time. To produce a high-quality question reaching both design objectives, we introduce a new multiround prompting approach. This method addresses multiple objectives in sequence across several rounds of prompting as shown in Figure 3. During the first round, the model is prompted to generate a question that targets the desired Bloom's cognitive level. In the second round, the model focuses on the specified knowledge point. Finally, the model combines both objectives into a single question, using the context from earlier rounds. To further improve the accuracy of the generated questions, we apply few-shot prompting by providing two examples in the final round of the prompt.



Basic Prompting:
Please generate a question that belongs to {*Bloom's level*} level of the Bloom's cognitive taxonomy, and it should involve knowledge points about derivatives.

Round Prompting with Two Shot:
Round 1:
Please generate a question that belongs to {*Bloom's level*} level of the Bloom's cognitive taxonomy.
Round 2:
Please generate a question that involves the knowledge points derivative.
Round 3:
Please generate a question that belongs to {*Bloom's level*} level of the Bloom's cognitive taxonomy, and it should involve knowledge points about derivatives.
Two example questions satisfying {*Bloom's level*} level and the knowledge point of derivative is shown below:
Question 1:{Text}
Question 2:{Text}

Figure 3. Round prompting with 2 shots.

3.2. Question Evaluation

In this study, the question evaluation model automatically assesses the quality of the generated questions instead of using a time-consuming evaluation process conducted by experts. Given a question, the evaluation model outputs its corresponding cognitive level. Figure 4 illustrates the overall process. The evaluation model verifies the alignment of each generated question with the target Bloom's cognitive level. If the question does not match the target level, the feedback from the evaluation model will instruct the generation model to retry question generation. The evaluation model is developed by fine-tuned ChatGLM2 on question—level pairs, reversing the format of the dataset used for the question generation model.

Additionally, the evaluation model checks for duplicates among the generated questions. A simple duplicate detection mechanism is implemented. If a newly generated question already exists in the selected question set, the generation model repeats the question generation step. Once a question satisfies Bloom's cognitive-level criteria and passes the duplication check, the evaluation model designates it as the final output.

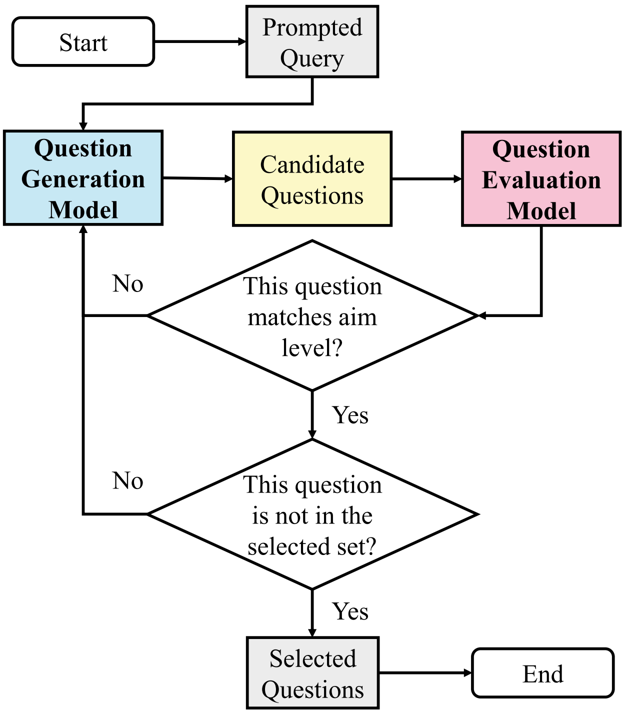

Start
Prompted Query
Question Generation Model
Candidate Questions
Question Evaluation Model
No
This question matches aim level?
Yes
No
This question is not in the selected set?
Yes
Selected Questions
End

Figure 4. The flowchart of the proposed method.

4. Experiments

This section provides a description of the evaluation process and an analysis of the experimental results. To demonstrate the impacts of various techniques on the improvement finally achieved by TwinStar, experimental data are displayed to show the performance gain achieved through multiple stages during the development of TwinStar, from the baseline model to a full-fledged dual-LLM engine. The experimental data are also provided to compare the performance of TwinStar against multiple state-of-the-art large language models, including GPT-3.5, GPT-4.0, Claude, and Bard. To evaluate the effectiveness of the proposed design scheme of a dual-LLM engine on various baseline models, TwinStar is also implemented using LLaMA2-13B. A similar trend of improvement is achieved as that shown by TwinStar implemented with ChatGLM2-6B. In terms of model selection in the last part of the experiments, we prefer a lightweight, open-source LLM, such as LLaMA2-13B, to represent models with different sizes and languages compared with ChatGLM2-6B.

All experiments were conducted on a workstation equipped with an NVIDIA A40 GPU (48 GB) (Nvidia, Santa Clara, CA, USA) and Python 3.8. The baseline model is ChatGLM2-6B, which contains approximately six billion parameters, consuming about 12 GB of memory.

4.1. Dataset

Dataset 1 and Dataset 2, shown in Table 1, are both used to fine-tune the question generation model and the cognitive-level evaluation model, similarly to in previous work [32,33]. These datasets were originally designed to recognize Bloom's cognitive levels using the original taxonomy [34]. The evaluation system in this work, however, adopts a revision of Bloom's taxonomy published in 2001 [21] that has been widely adopted in the fields of teaching and assessing. The revised Bloom's model consists of six cognitive levels, i.e., remembering, understanding, applying, analyzing, evaluating, and creating, from low- to high-order thinking. Both datasets used in our experiments were relabeled to keep them consistent with the revised Bloom's taxonomy. The distribution of the datasets is shown in

Table 1. Each dataset contains an balanced distribution of 100 questions per category to ensure the stability of the experimental results for this study.

Table 1. Distribution of applied datasets.

Bloom’s Level	Dataset 1	Dataset 2	Sum
Remembering	26	100	129
Understanding	23	100	123
Applying	15	100	115
Analyzing	23	100	123
Evaluating	24	100	124
Creating	30	100	130
Sum	141	600	741

The sample questions from both datasets adopt a format of question–answer pairs and cover a broad spectrum of subjects, including literature, mathematics, biology, medicine, geography, and computer science. The target subject of the question generation model in this work is university-level calculus, including functions, derivatives, integrals, and related concepts.

The experiments evaluate whether the generated questions align with the design objectives in terms of Bloom’s cognitive levels and knowledge points. Two metrics were employed to measure the accuracy of the generated questions: Knowledge relevance (KR) measures whether a question pertains to the intended knowledge domain, and Bloom adherence (BA) assesses whether a question meets the target Bloom’s level. These metrics have also been applied in related studies [31]. There are 20 questions generated for each Bloom’s level, resulting in 120 questions in total for evaluating the various approaches used in our experiments.

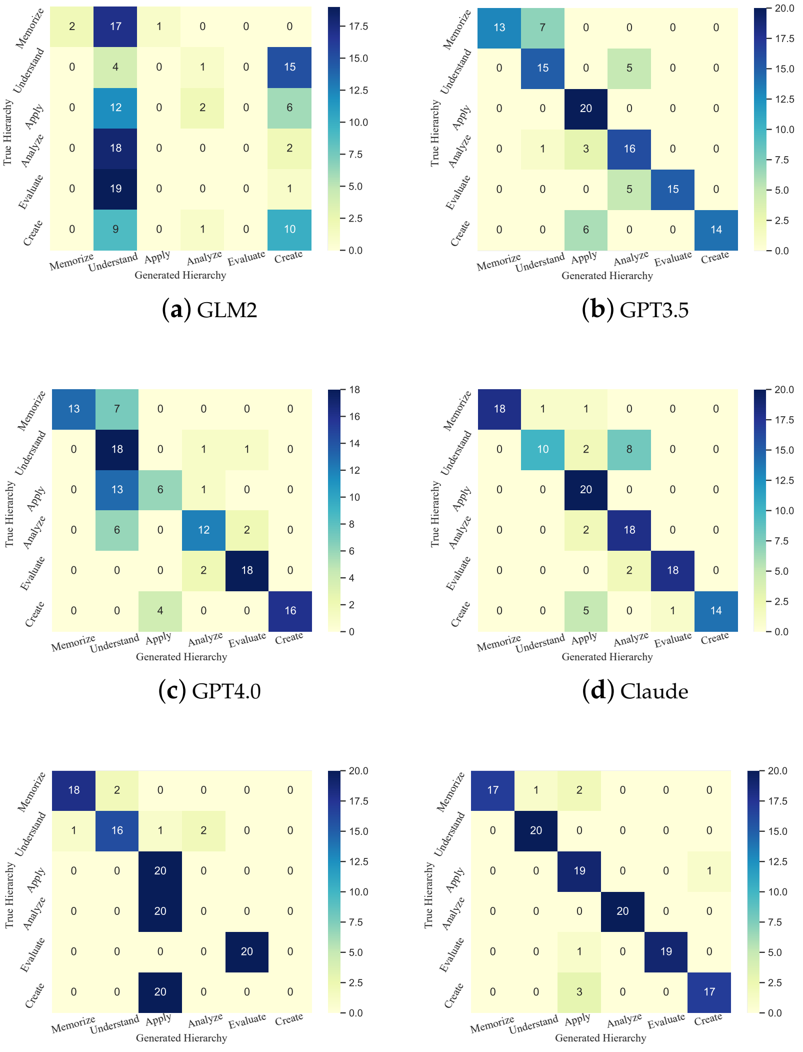
4.2. Results

The performance of the proposed framework for TwinStar, a dual-LLM engine for enhanced test question generation, is evaluated and analyzed in this subsection. The framework orchestrates two LLMs: a question generation model and a cognitive-level evaluation model. Our implementation of TwinStar, using two fine-tuned, lightweight baseline models (ChatGLM2-6B) and prompting technology, generates test questions targeting specific knowledge points in university-level calculus at designated cognitive levels without an additional question bank or expert evaluation. The experimental results show that the proposed design framework, TwinStar, outperforms the state-of-the-art LLMs, including GPT-3.5, GPT-4, Claude, and Bard, for effective test question generation in terms of cognitive-level adherence and knowledge relevance. To show the effectiveness of the TwinStar framework, we also implemented TwinStar using LLaMA2-13B. The experimental data show a similar trend of improvement.

4.2.1. Recognizing Bloom’s Cognitive Levels Through the Evaluation Model

In order to improve the effectiveness of the dual-LLM engine in recognizing Bloom’s cognitive levels, we fine-tuned the cognitive-level evaluation model using 1000 training questions. Then, we assessed its performance on a test set consisting of 141 questions. As a result, the accuracy of cognitive-level recognition was improved from 40% to 70%.

The accuracy of cognitive-level recognition using various LLMs is illustrated by heat maps, as shown in Figure 5a. The x-axis represents the target cognitive levels and the y-axis represent cognitive levels that have been confirmed by LLMs. A dark-colored cell indicates that the recognized level matched with the target cognitive level of a test question for most of the experiment samples. Figure 5a shows that TwinStar achieved the highest score in the test of cognitive levels compared with stand-alone LLMs, including GLM2, GPT-4, GPT-3.5, Claude, and Bard.



(**a**) GLM2 (**b**) GPT3.5

(**c**) GPT4.0 (**d**) Claude

(**e**) Bard (**f**) Twinstar

Figure 5. Heat maps showing the effectiveness of cognitive-level recognition using various LLMs.

4.2.2. Evaluating Quality of Test Question Generation

Table 2 shows the evaluation results of both performance metrics, knowledge relevance (KR) and Bloom's cognitive-level adherence (BA). Table 2 also shows the average accuracy achieved with equally weighed KR and BA, as well as the standard deviation (SD), indicating the balance of performance improvement in terms of both KR and BA measurements across all the evaluated models. To demonstrate the effectiveness of TwinStar, we present the improvement in performance metrics achieved in several steps using various techniques. As shown in Table 2, the baseline model, ChatGLM2 (GLM2), was the least effective model in aligning the generated questions with the target Bloom's cognitive levels and knowledge points. After fine-tuning with Bloom's instruction–question pairs, GLM2-pt shows a significant rise in BA, while the KR value remains low. A low score for KR reflects the fact that the training dataset used in this step is different from the calculus domain. The value of KR is constantly improved when advanced optimization techniques are applied in multiple steps. The application of multiround prompting (GLM2-pt-RP) noticeably boosts KR while slightly reducing BA. The resultant measurement indicates that multiround prompting effectively guides the model toward the target knowledge domain.

After complementing two additional examples in the prompt in GLM2-pt-RP-2shot, the KR measurement was further improved. The fine-tuned GLM2 model can then effectively recognize the relationship between questions and their respective cognitive levels. On the other hand, large LLMs, inluding GPT-3.5, GPT-4.0, Claude, and Bard, all exhibit relatively low performance in recognizing appropriate Bloom's cognitive levels, resulting in unbalanced performance in terms of cognitive-level adherence and knowledge point relevance. Finally, with the help of the evaluation model, TwinStar achieves significant gains across all metrics: BA, KR, and overall average improvement by factors of 6, 29, and 10, respectively, relative to the baseline (GLM2).

Table 2. Performance in question generation for different models.

Model	BA	KR	Average	SD
GLM2	0.13	0.03	0.08	0.05
GLM2-pt	0.90	0.23	0.57	0.36
GLM2-pt-RP	0.86	0.64	0.75	0.11
GLM2-pt-RP-2shot	0.83	0.73	0.78	0.05
GPT-3.5	0.78	1.00	0.89	0.11
GPT-4.0	0.77	1.00	0.89	0.12
Claude	0.82	0.89	0.86	0.04
Bard	0.62	1.00	0.81	0.19
TwinStar	0.93	0.89	0.91	0.02

A comparison between TwinStar and other large language models reveals significant improvement in the overall performance achieved by the proposed TwinStar design. Notably, TwinStar achieves the highest Bloom adherence (BA) among all models, outperforming GPT-3.5, GPT-4.0, Claude, and Bard (renamed as Gemini) by 19%, 21%, 13%, and 50%, respectively, which underscores the effectiveness of the generation evaluation framework. Figure 5 provides heat maps illustrating how each model generates questions aligning with the six Bloom's cognitive levels, reflecting model-specific performance across Bloom's categories. In terms of knowledge relevance (KR), TwinStar and Claude obtain the same result, whereas GPT-3.5, GPT-4.0, and Bard attain the highest KR values. With respect to overall performance (average), TwinStar exceeds GPT-3.5, GPT-4.0, *Claude*, and *Bard* by 2%, 2%, 6%, and 12%, respectively.

Note that the TwinStar architecture consists of two lightweight ChatGLM2 models containing just 12 billion parameters in total, a fraction of the parameters of large LLMs, such as GPT-3.5. In other words, TwinStar achieves better accuracy with far fewer parameters. Therefore, it can produce the desired test questions with a much smaller memory space requirement. This improvement is achieved through a carefully designed dual-model architecture, as well as with the assistance of fine-tuning and prompting techniques. This finding also indicates that well-designed, small language models can outperform their large counterparts, dramatically reducing the costs of model building and management. The proposed approach could be a valuable direction for exploring the efficient deployment of AI engines on resource-constrained computing devices.

To show the effectiveness of the proposed technique on other baseline models, we selected LLaMA2-13B, another open-source LLM containing 13 billion parameters and consuming about 26 GB of memory space, to implement TwinStar and evaluate its performance. There are more than twice as many parameters in LLaMA2 than in GLM2, while it remains a lightweight model compared with LLaMA3, GPT-4, Bard, etc.

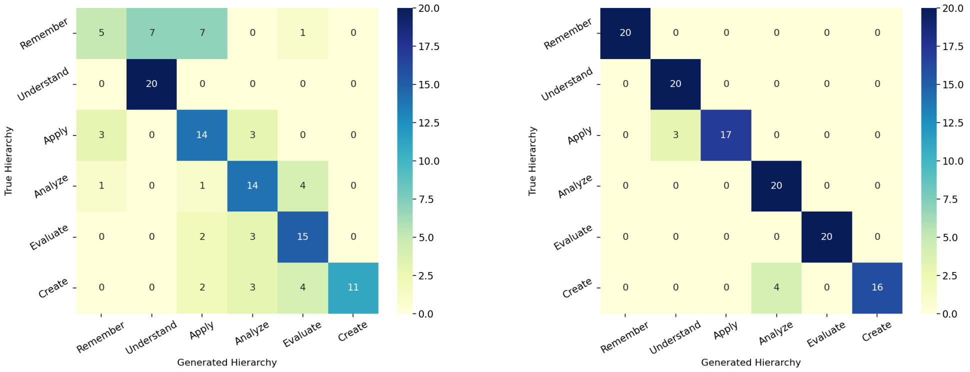
Table 3 presents the results of applying various prompting techniques step-by-step for LLaMA2-13B. Initially, LLaMA2 shows a performance advantage over GLM2, owing to its number of parameters.

After applying multiround prompting, both the BA and KR of LLaMA2-pt-RP show notable improvements. However, the quality of the generated questions slightly decreases when two examples are provided in LLaMA2-pt-RP-2shot. Table 3 shows both the average accuracy achieved with equally weighed KR and BA and the standard deviation (SD), indicating the balance of performance improvement in terms of knowledge reference (KR) and Bloom's cognitive-level adherence (BA). We observe that the sentences in the questions generated by LLaMA2 tend to be more complex compared to those generated by GLM2. In this case, the example questions, which are relatively simple, are not sufficient to effectively guide LLaMA2 as prompts.

Table 3. Performance of question generation using the proposed methods based on LLaMA2-13B.

Model	BA	KR	Average	SD
LLaMA2	0.66	0.78	0.72	0.06
LLaMA2-pt	0.68	0.72	0.70	0.02
LLaMA2-pt-RP	0.76	0.88	0.82	0.06
LLaMA2-pt-RP-2shot	0.74	0.67	0.70	0.04
TwinStar (LLaMA2)	0.94	0.83	0.89	0.05

Figure 6 illustrates heat maps of the cognitive levels for both LLaMA2-13B and TwinStar (LLaMA2). The results clearly show that our TwinStar framework also significantly enhances the BA measurement for LLaMA2-13B. It is important to note that the number of parameters in LLaMA2-13B is considerably lower than that in GPT-3.5 and GPT-4.0. The results show that our proposed technique, TwinStar, can also improve question generation quality for large-size language models.



(**a**) LLaMA2-13B (**b**) TwinStar (LLaMA2)

Figure 6. Bloom levels for LLaMA2-13B and the version with the applied TwinStar technique, named TwinStar (LLaMA2).

5. Conclusions and Future Work

Maximizing the capability of AI agents with minimal resource consumption is a trend of future development of AI. In this paper, we try to address the problem of applying LLMs in a professional domain with specified requirements, in particular, generating cognitive-level-aware test questions for educational purposes and stimulating interest in finding new solutions to effectively upgrade the intelligence of LLMs with minimal resource consumption. A new design framework, TwinStar, is proposed, featuring a dual-LLM engine and incorporating a set of fine-tuning and prompting techniques to enhance the capabilities of the AI-assisted test question generation system. The core of TwinStar is a pair of fine-tuned instances of ChatGLM2-6B, which is a lightweight LLM containing just 6 billion parameters and consuming about 12 GB memory. The experimental results show that TwinStar outperforms state-of-the-art LLMs in terms of accuracy, memory efficiency, and overall performance, with a fraction of the resources consumed by state-of-the-art LLMs, including GPT-4, Claude, Bard, etc. This study inspires us to further investigate interesting problems, such as conducting in-depth studies on the impacts of reinforcement learning and modal distillation on the application of LLMs in professional domains and exploring the possibilities of using a multiple-agent AI engine to obtain collective intelligence on a network of mobile devices.

Author Contributions: Conceptualization, Q.Z., H.W. and X.C.; methodology, Q.Z. and H.W.; validation, H.W. and X.C.; formal analysis, H.W.; investigation, H.W.; data curation, H.W. and X.C.; writing—original draft preparation, H.W.; writing—review and editing, Q.Z. and H.W.; visualization, H.W.; supervision, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by NSFC 62372183 and Shanghai STCSM 24LZ1400500.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://figshare.com/articles/dataset/Exam_Question_Datasets/22597957].

Acknowledgments: This work is partially supported by NSFC 62372183 and STCSM 24LZ1400500.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. OpenAI. *ChatGPT: Optimizing Language Models for Dialogue*; Technical Report; OpenAI: San Francisco, CA, USA, 2022.
2. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
3. Google DeepMind. *Gemini Technical Report: A Family of Highly Capable Multimodal Models*; Technical Report; Google: Mountain View, CA, USA, 2023.
4. Anthropic. *The Claude 3 Model Family: Opus, Sonnet, Haiku*; Technical Report; Anthropic: San Francisco, CA, USA, 2024.
5. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
6. Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. Glm-130b: An open bilingual pre-trained model. *arXiv* **2022**, arXiv:2210.02414.
7. Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; Tang, J. GLM: General Language Model Pretraining with Autoregressive Blank Infifilling. *arXiv* **2022**, arXiv:2103.10360.
8. Jiao, W.; Wang, W.; Huang, J.T.; Wang, X.; Tu, Z. Is ChatGPT a good translator? A preliminary study. *arXiv* **2023**, arXiv:2301.08745.
9. Laskar, M.T.R.; Bari, M.S.; Rahman, M.; Bhuiyan, M.A.H.; Joty, S.; Huang, J.X. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 431–469.
10. Wang, J.; Liang, Y.; Meng, F.; Shi, H.; Li, Z.; Xu, J.; Qu, J.; Zhou, J. Is chatgpt a good nlg evaluator? A preliminary study. *arXiv* **2023**, arXiv:2303.04048.
11. Nov, O.; Singh, N.; Mann, D. Putting ChatGPT's medical advice to the (Turing) test: Survey study. *JMIR Med. Educ.* **2023**, *9*, e46939. [CrossRef] [PubMed]
12. Chen, S.; Kann, B.H.; Foote, M.B.; Aerts, H.J.; Savova, G.K.; Mak, R.H.; Bitterman, D.S. The utility of ChatGPT for cancer treatment information. *MedrXiv* **2023**, 2003–2023. [CrossRef]
13. Blair-Stanek, A.; Holzenberger, N.; Van Durme, B. Can GPT-3 perform statutory reasoning? In Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, Braga, Portugal, 19–23 June 2023; pp. 22–31.
14. Yu, F.; Quartey, L.; Schilder, F. Legal prompting: Teaching a language model to think like a lawyer. *arXiv* **2022**, arXiv:2212.01326.
15. Koubaa, A.; Boulila, W.; Ghouti, L.; Alzahem, A.; Latif, S. Exploring ChatGPT capabilities and limitations: A critical review of the nlp game changer. *Preprints* **2023**, *2023030438*, 1–29. [CrossRef]

16. Malinka, K.; Peresíni, M.; Firc, A.; Hujnák, O.; Janus, F. On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree? In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, Turku, Finland, 10–12 July 2023; pp. 47–53.
17. Susnjak, T. ChatGPT: The end of online exam integrity? *arXiv* **2022**, arXiv:2212.09292.
18. Bordt, S.; von Luxburg, U. Chatgpt participates in a computer science exam. *arXiv* **2023**, arXiv:2303.09461.
19. Qu, Z.; Yin, L.; Yu, Z.; Wang, W. CourseGPT-zh: An educational large language model based on knowledge distillation incorporating prompt optimization. *arXiv* **2024**, arXiv:2405.04781v1.
20. Muse, H.; Bulathwela, S.; Yilmaz, E. Pre-training with scientific text improves educational question generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 16288–16289.
21. Anderson, L.W.; Krathwohl, D.R. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*; Longman: New York, NY, USA, 2001.
22. Singh, R.; Gulwani, S.; Rajamani, S. Automatically generating algebra problems. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canda, 22–26 July 2012; Volume 26, pp. 1620–1628.
23. Andersen, E.; Gulwani, S.; Popovic, Z. A trace-based framework for analyzing and synthesizing educational progressions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 27 April–2 May 2013; pp. 773–782.
24. Wang, H.; Zhuge, Q.; Sha, H.-M.S.; Xia, J.; Xu, R. Exploring Multiple-Objective Optimization for Efficient and Effective Test Paper Design with Dynamic Programming Guided Genetic Algorithm. *Math. Biosci. Eng.* **2024**, *21*, 3668–3694. [CrossRef] [PubMed]
25. Zhou, Q.; Huang, D. Towards generating math word problems from equations and topics. In Proceedings of the 12th International Conference on Natural Language Generation, Tokyo, Japan, 29 October 29–1 November 2019; pp. 494–503.
26. Liyanage, V.; Ranathunga, S. Multi-lingual mathematical word problem generation using long short term memory networks with enhanced input features. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 4709–4716.
27. Xia, J.; Wang, H.; Zhuge, Q.; Sha, E. Knowledge Tracing Model and Student Profile Based on Clustering-Neural-Network. *Appl. Sci.* **2023**, *13*, 5220. [CrossRef]
28. Williams, S. Generating mathematical word problems. In Proceedings of the 2011 AAAI Fall Symposium Series, Arlington, VA, USA, 4–6 November 2011.
29. Ahmed, U.Z.; Gulwani, S.; Karkare, A. Automatically generating problems and solutions for natural deduction. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 1968–1975.
30. Drori, I.; Zhang, S.; Shuttleworth, R.; Tang, L.; Lu, A.; Ke, E.; Liu, K.; Chen, L.; Tran, S.; Cheng, N.; et al. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2123433119. [CrossRef] [PubMed]
31. Elkins, S.; Kochmar, E.; Serban, I.; Cheung, J.C. How Useful are Educational Questions Generated by Large Language Models? In Proceedings of the International Conference on Artificial Intelligence in Education, Kobe, Japan, 20–22 March 2023; pp. 536–542.
32. Yahya, A.A.; Toukal, Z.; Osman, A. Bloom's taxonomy–based classification for item bank questions using support vector machines. In *Modern Advances in Intelligent Systems and Tools*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 135–140.
33. Mohammed, M.; Omar, N. Question classification based on bloom's taxonomy using enhanced tf-idf. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2018**, *8*, 1679–1685. [CrossRef]
34. Bloom, B.S.; Engelhart, M.D.; Furst, E.J.; Hill, W.H.; Krathwohl, D.R. *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook 1: Cognitive Domain*; Longman: New York, NY, USA, 1956.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.