Using Large Multimodal Models to Extract Knowledge Components for Knowledge Tracing from Multimedia Question Information

Hyeongdon Moon
EPFL
donim@andrew.cmu.edu

Richard Lee Davis
KTH Royal Institute of
Technology
rldavis@kth.se

Seyed Parsa Neshaei
EPFL
seyed.neshaei@epfl.ch

Pierre Dillenbourg
EPFL
pierre.dillenbourg@epfl.ch

ABSTRACT

Knowledge tracing models have enabled a range of intelligent tutoring systems to provide feedback to students. However, existing methods for knowledge tracing in learning sciences are predominantly reliant on statistical data and instructor-defined knowledge components, making it challenging to integrate AI-generated educational content with traditional established methods. We propose a method for automatically extracting knowledge components from educational content using instruction-tuned large multimodal models. We validate this approach by comprehensively evaluating it against knowledge tracing benchmarks in five domains. Our results indicate that the automatically extracted knowledge components can effectively replace human-tagged labels, offering a promising direction for enhancing intelligent tutoring systems in limited-data scenarios, achieving more explainable assessments in educational settings, and laying the groundwork for automated assessment. [1]

Keywords

Knowledge Tracing, Multimodal Models, Multimedia Question Information, Knowledge Components

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) are advanced computer programs that provide personalized and adaptive educational instruction to learners. For over half a century, they have been a subject of active research and discussion in the interdisciplinary field of education and artificial intelligence [28]. These systems integrate techniques from artificial intelligence to deliver tailored instruction, dynamically adjusting

---

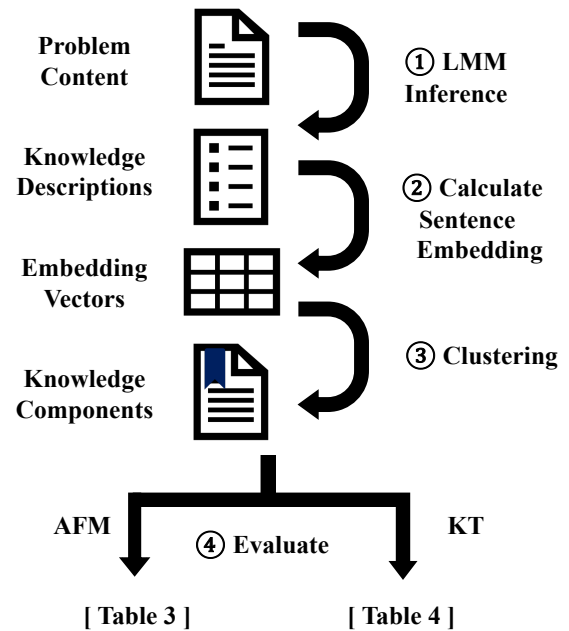[1] Codes are available at https://github.com/DoniMoon/LLMKT

Figure 1: Overview of our experiment. We evaluate the quality of extracted knowledge components by utilizing them in two validation tasks, Knowledge Tracing (KT) and Additive Factor Model (AFM).

to the needs and progress of individual students. By simulating one-on-one tutoring, ITS offer immediate and specific feedback, enhancing student engagement and improving learning outcomes. This combination of real-time adaptability and personalized feedback makes ITS a valuable tool in modern education, bridging the gap between traditional classroom methods and individualized learning approaches [31].

Knowledge Tracing (KT) is a foundational task in Intelligent Tutoring Systems (ITS), aiming to model a student's knowledge state and predict future performance on educational tasks. Traditional KT models rely heavily on statistical techniques to analyze historical problem-solving data. Early approaches, such as Bayesian Knowledge Tracing (BKT),

used hidden Markov models to estimate the probability of a student knowing a particular skill at any given time [7].

More recent models have leveraged advancements in machine learning, particularly deep learning, to enhance predictive accuracy [1]. Deep Knowledge Tracing (DKT) was one of the first models to apply recurrent neural networks (RNNs) to KT, demonstrating significant improvements over BKT by capturing the sequential nature of students' learning processes. Subsequent models such as Self-Attentive Knowledge Tracing (SAKT) and Separated Self-AttentIve Neural Knowledge Tracing (SAINT) have further refined these approaches [5, 32].

Knowledge Components (KCs) are defined within the Knowledge-Learning-Instruction framework as *acquired units of cognitive function or structure that can be inferred from performance on a set of related tasks* [19]. While we cannot directly observe the changes in a student's KCs, we infer them through interactions during assessment and instruction. In Knowledge Tracing, each problem is often labeled with its corresponding KCs, which are typically assigned by human experts based on their expected relevance to the problem and statistically validated for their ability to explain student performance.

There are several methods to define these KCs. [35] introduces four categories of domain modeling: 'KCs as disjoint sets', 'Multiple KCs per item', 'Hierarchy of KCs', and 'KCs with prerequisites'. In the KT domain, the most common approaches are modeling KCs as disjoint sets or mapping multiple KCs to an item. The simplest approach is to map each question to a single KC, as used in Bayesian KT, SAKT, and SAINT. When mapping multiple KCs, the relationship between KCs and items is represented in a Q-matrix, where rows represent questions and columns represent KCs. Cognitive Diagnosis Models (CDM) like DINA, NIDA, and generalized DINA use this Q-matrix for KT [10], thus supporting multiple KCs labeled for each question.

Well-defined KCs are essential for accurately predicting learner performance, building effective adaptive learning systems, and providing efficient learning support and improvement. Due to this importance, research has been conducted to improve KC models through Difficulty Factors Assessment (DFA) and to evaluate the improvements using the Additive Factors Model (AFM), ultimately creating better learner models [23, 18].

Our research proposes a novel approach leveraging the instruction-tuned Large Multimodal Model (LMM) to extract and utilize Knowledge Components (KCs) from educational content. Figure 1 illustrates the overall architecture of our approach. Our method involves parsing educational materials to extract text and images, using the GPT-4o API to identify and describe inherent KCs, and clustering similar components based on sentence embeddings. This automation not only improves KC extraction but also enhances the prediction of student performance on new and unseen content.

We observed performance improvements when using LMM-generated KCs as additional features in various KT methods. For example, in the Performance Factors Analysis (PFA) method, LMM-generated KCs resulted in greater performance increases compared to human-generated KCs. Other KT methods also showed comparable performance improvements. Additionally, we analyzed the performance factors of generated KCs using the AFM.

We compared LMM-generated KCs to human-generated KCs by using them in four different knowledge tracing models across six datasets. Overall, when using LMM-generated KCs, we demonstrated comparable or superior performance to human-generated KCs. To encourage further research in this area, we have refined and publicly released KT benchmarks with content data across five domains. By providing these benchmarks, we aim to facilitate the development of more advanced KT methodologies that can fully utilize the potential of LMM-generated KCs.

In summary, our contributions are three-fold:

1. Introducing a novel zero-shot KC generation methodology that can be applied to general domains and diverse modalities supported by LMMs.

2. Our automatically generated KCs model students' problem-solving data as effectively as human-created KCs in both KT and AFM.

3. We publish a reproducible KT benchmark with parsed content data, advancing content-aware Knowledge Tracing methods.

2. RELATED WORK
2.1 Utilizing LLMs to improve ITS and Knowledge Tracing

The emergence of advanced NLP tools in recent years, especially instruction-tuned LLMs like ChatGPT, has significantly enhanced ITS by providing natural, human-like interactions [22, 13]. Large language models (LLMs) have been shown to provide support in a range of areas, including, but not limited to, planning learning instruction [16], scaffolding [12, 25], and helping students solve math word problems [15]. They assist learners in multiple ways, such as participating in back-and-forth instructional conversations [21, 38] or providing feedback to students [9, 14].

Moreover, LLMs have also been used to improve the accuracy and performance of the KT models in the backbone of ITS. LM-KT was proposed to perform KT even when there is no prior problem-solving data from the student. The LM-KT model trains GPT-2 to perform Knowledge Tracing on content without any prior student interaction records [41]. In a basic second-language acquisition problem space, LM-KT models student success rates based on the input of natural language sentences as questions. Following LM-KT, other work has shown that leveraging the generalizability of LLMs can enhance the performance of Knowledge Tracing [29, 45] and address cold-start problems [20]. However, their applicability to other domains is limited, as the language model needs to be fine-tuned on the specific domain.

LLMs have the ability to digest information from source documents which can help with extracting information from
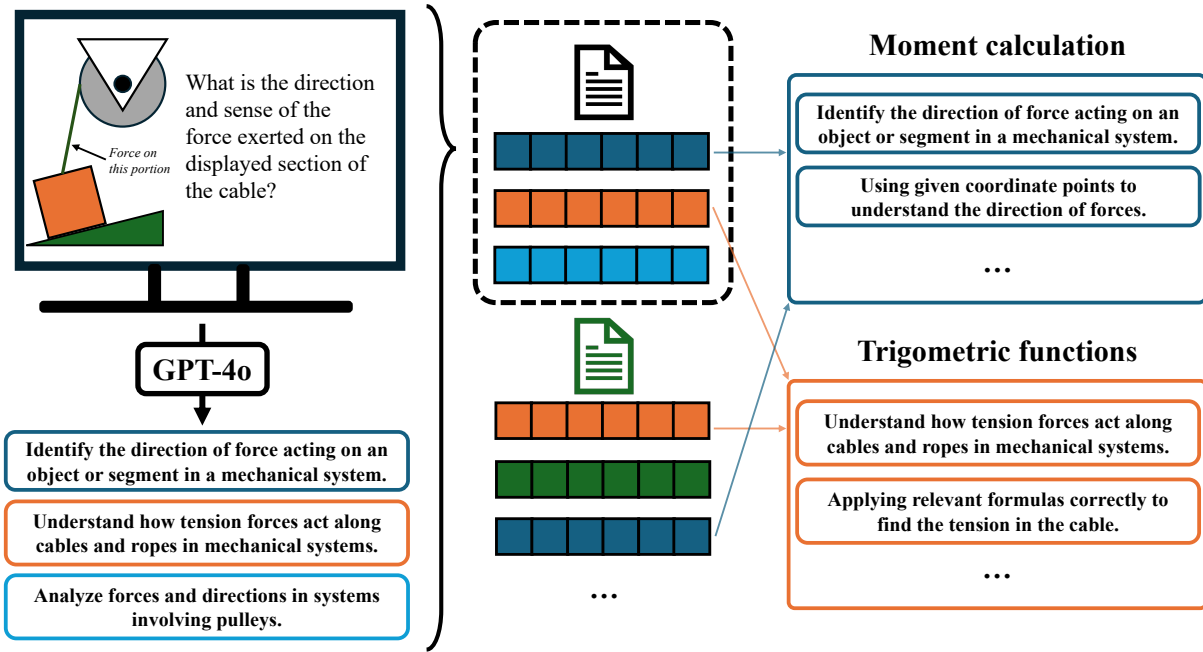
Figure 2: Overall procedure of extracting knowledge components. First, we extract knowledge components using GPT-4o API from the parsed question content. Then, we calculate sentence embedding for all descriptions of the generated knowledge components. At last, we group each knowledge components into similar semantic group using a clustering approach.

large-scale educational data; for example, they have been used for summarize content by intention [40] or extracting key points out of educational data [8]. However, the application of LLMs to improve the performance of KT models, specifically by assisting in the process of KC extraction, has been rare in the literature. We address this gap by providing and evaluating a structured KC extraction method using LLMs in the loop, with the goal of improving the accuracy of KC models while reducing the need for human labeling labor.

2.2 Extracting Knowledge Components

Commonly, the assignment of KCs to questions is done by human experts, e.g., instructors [42]; however, this requires manual labor, making it less suitable for course offerings with a high number of questions. As a result, previous researchers have explored methods to move from expert-annotated KCs towards extracting KCs from the problem information automatically, which can aid in developing more accurate KT models. For example, a methodology to extract KCs from documents using classical NLP techniques and annotate these documents for application in adaptive online textbooks has been proposed [43]. Additionally, there is a study that validated the accuracy of a method for automatically generating skills for problems by fine-tuning a language model on problem and skill label data to enable computer adaptive tests [42].

Unlike previous studies, we perform the task of tagging the Knowledge Components (KCs) required by questions without using pre-trained problem data or provided schemas of knowledge components. Additionally, we focus on generating entities that clearly correspond to the previous concept of 'knowledge component', rather than simply using terms like skill, knowledge, or tag, and we conduct validation for

the KCs. The study most closely related to ours analyzed the relation of questions and KCs using LLMs [27]. This study uses pre-trained LLMs to verify that if a problem generated from a single KC is a good one, there is a strong dependency on that KC, for the purpose of evaluating automatically generated questions. Conversely, we aim to extract KCs directly from the questions, under the assumption that each question is designed to test specific knowledge.

3. METHODOLOGY
As illustrated in Figure 2, our overall pipeline involves preprocessing student interaction data enriched with content information, extracting KCs, evaluating the quality of the extracted KCs, and analyzing their utility across various KT methods.

3.1 Dataset
We chose to process the OLI datasets from CMU Datashop[2] since the OLI_Statics2011 dataset is a well-known benchmark for knowledge tracing which has publicly available content data[18, 1]. The Open Learning Initiative (OLI) project provides research-based courseware suitable for various class formats and supports advanced research. We gained access to several domains of OLI learning content from CMU Datashop. The preprocessing code for the datasets is publicly available, and to facilitate reproduction while protecting the content data, the parsed files for each dataset have been uploaded to the corresponding entries in Datashop.

From CMU Datashop, all transaction data was extracted by selecting "all data" in the "Export" tab and choosing "By Transaction" in the detailed options. The content data was collectively downloaded by clicking the "Download Problem
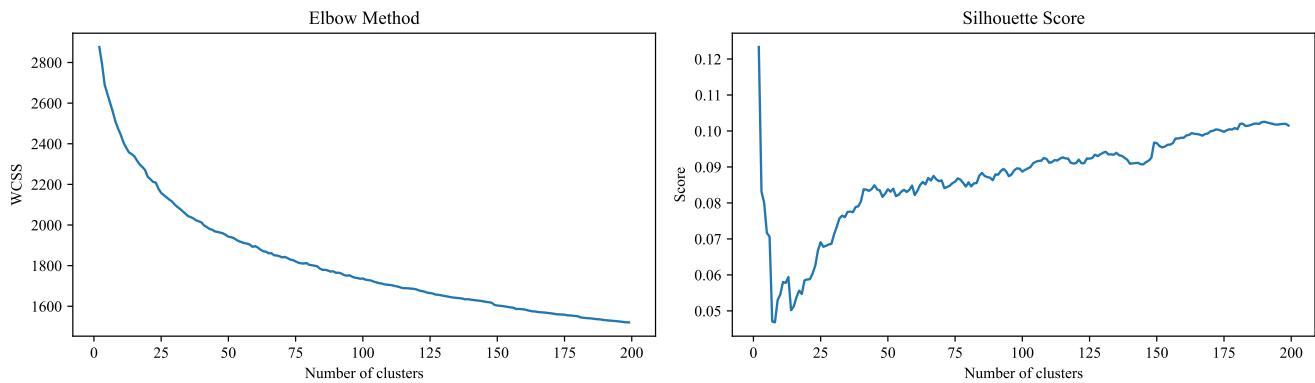
---

[2]https://pslcdatashop.web.cmu.edu/

Figure 3: Clustering Scores used to determine the number of clusters in oli_statics. The left graph (Elbow Method) shows the WCSS (y-axis) values for different numbers of clusters (x-axis), indicating that WCSS gradually decreases as the number of clusters increases. The right graph (Silhouette Score) presents the silhouette scores measured over the same range of cluster numbers, generally showing that the score tends to rise and fluctuate as the number of clusters increases.

Content" button in the "Dataset Info/Problem List" tab. We used the swftools package [3] in the Windows OS environment to extract images from the SWF file, which is not supported anymore due to the deprecation of Flash support.

all mp3 files are converted into text files, using whisper-large-v2 [36].

3.1.1 OLI Engineering Statics

For the Statics dataset, we used the Fall 2011 version[4]. While other versions included a wider variety of KC models, the number of users and transactions was smaller compared to other statics benchmarks. Therefore, we considered the Fall 2011 version to be the superset and used it. The KC model named F2011 was used as the human tag. The original 361,092 transactions were reduced by about half to 189,047.

3.1.2 OLI Principles of Computing

The content of the OLI computing dataset [5] was different from other subjects, as it contained content from multiple subjects under the root directory. To maintain consistency with the method used for processing other datasets, only the content from Principles_of_Computing_v_1_10, which includes the version in the subject name, was used. The KC models `principles_of_computing_1_13` and `principles_of_computing_1_10` were used to calculate AFM scores. The transaction data was filtered to include only those containing this content. As a result, the original transaction dataset, which had 37,233 rows, was reduced to 16,951 rows.

3.1.3 OLI French

We used the 'French1 - Spring 2014' dataset for the French dataset[6]. Since it included various KC models, we used *Bonnie's Model*, which had the best AFM performance. To train

---

[3] http://www.swftools.org/
[4] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507
[5] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1806
[6] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=918

AFM, *Level4* KC model, which also reports good AFM performance, is used. This dataset contained many items with voice mp3 files, so we converted these files to text using the whisper-large-v2 model and inserted the transcriptions into the question text. At the position in the HTML where the mp3 execution was embedded, we prefixed the converted text with '[transcription of embedded mp3 file]:'. Out of the total 278,489 rows of transactions, the final remaining data consisted of 53,255 rows.

3.1.4 OLI Biology

We used the 'Oli_biology' dataset from the 'Bridge to Success' project for the Biology data[7]. The KC model named *intro_biology-1_0* was used as the human label. After processing, 3,285,685 rows remained out of the original 5,852,795 rows of transactions.

3.1.5 OLI Psychology

We used the 'Psychology MOOC GT - Spring 2013' dataset for the Psychology data[8] and utilized the KC model named *psychology-1-4*. After processing, 1,935,496 rows remained out of the original 2,493,609 rows of transactions.

3.2 Knowledge Component Extraction

To leverage the capabilities of instruction-tuned LLMs for educational purposes, we first focus on extracting knowledge components from educational content. Our approach involves the following steps:

3.2.1 Content Processing

The content data available from the CMU DataShop consists of HTML pages representing the learning materials that students interact with [18]. We parsed this data, extracting text and images from Flash files embedded within the content, and converted MP3 files into text data using the *whisper-large-v2* model [36]. We processed the image files to embed them in the Chat Template for the OpenAI

---

[7] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1148
[8] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=863

345

Understanding how to calculate the x and y components of forces.
Applying trigonometric functions to resolve forces into perpendicular components.
Understand how to decompose a force into its perpendicular components.
Calculate the position vector from point O to the point of application of each force.
Learn how to represent forces using vectors.
Decomposing a given force into its x and y components based on angles provided.
Decompose a force into its components.
Using trigonometric functions to resolve forces into components.
Calculate the x and y components of a vector given its magnitude and direction.
Use sine and cosine functions to resolve forces into their components.

Understand how to determine sense and direction of a force.
Assigning labels to the identified forces based on their origin and point of interaction.
The direction and point of application of a force determine how it contributes to the equilibrium conditions.
Recognize that forces can act in multiple directions.
Understanding the appropriate direction of force at a given point.
Determine the resultant direction of the applied force from one body to another.
Determine the force labels in the context of the question.
Understanding the direction and magnitude of vertical forces in a system.
The vector represents the magnitude and direction of a force.
Predicting the directions of forces exerted at a joint based on given resultant forces.

Figure 4: Example of the knowledge components belonging to the cluster **Trigonometric Relationships** (above) and **Moment Calculation** (below) in *Statics* dataset.

API [9], preserving their positions and order from the original HTML content where the images were located. Each problem was matched with the corresponding steps in the CMU DataShop transaction data, creating datasets for five subjects: Statics, Psychology, Biology, Computing, and French.

3.2.2 Knowledge Component Extraction Prompt
Using the OpenAI API for the GPT-4o model, we extracted the knowledge components for each problem as shown in Figure 2. Each problem can have multiple knowledge components, with each component consisting of a name field (1–3 words) and a description field (1–2 sentences). We applied this basic zero-shot prompt from Fig. 2 only once and, being satisfied with the qualitative output, did not undertake any further prompt tuning.

3.2.3 Clustering Knowledge Components
The extracted KCs consisted of natural language sentences, which occasionally referred to the same topic but with different sets of words. To utilize these components for problem correlation and be able to assign the same identifiers to semantically similar KCs, we needed to group them together. To do this, we computed sentence embeddings for each component and performed clustering based on similarity. The optimal number of clusters was determined by maximizing the silhouette score of the clustering [37]. We compared the performance of the Sentence-T5-XXL model [30] and OpenAI's *text-embedding-3-large* model for this task [10].

Figure 3 shows the WCSS and Silhouette Score of the K-means clustering method as the number of clusters varies from 2 to 200 in the oli_statics dataset. Due to the instability of the Silhouette Score when the number of clusters

is very small, we analyzed cases where the number of clusters is greater than 10. Each point in Figure 6 represents the local maximum Silhouette Score within one of the 10 bins divided between the range of 10 to 200 clusters, and the AFM performance was measured using these cluster numbers.

Meanwhile, for zero-shot implementation, using more than 100 KCs, which maximizes the Silhouette Score, made it impossible to have at least one problem with each KC in both the train and test splits. Therefore, we used the number of clusters at the local maximum of the third bin, where such splits were feasible for all datasets. The selected numbers of clusters were 52, 63, 52, 61, and 49 for computing, statics, French, psychology, and biology, respectively.

3.3 Knowledge Component Quality Evaluation
To validate the effectiveness of our knowledge component extraction method, we conducted a comprehensive quality evaluation across five different datasets. Using the Additive Factors Model (AFM), we measured the Root Mean Square Error (RMSE) and compared it with the RMSE of human-generated KC mappings for each dataset. This evaluation provided a robust assessment of the accuracy and reliability of the LLM-extracted knowledge components.

The KCs used in this validation were evaluated by measuring the silhouette score of K-means clustering from 2 to 200 clusters. We automatically determined the optimal number of clusters by detecting the knee point of the silhouette score change [37, 39]. To further verify the quality of KCs based on the level of clustering, we divided the entire range of 2 to 200 clusters into ten sections. For each section, we identified the point where the silhouette score was at its local maximum and observed the performance change of the AFM when KCs were generated with the corresponding number of clusters.

RMSE was measured not only for the entire dataset but also in environments where student ID information and item ID

---

information were masked, respectively. This ablation experiment was done to identify which biases the model was utilized to demonstrate performance. For performance measurement, we utilized PyAFM, a Python implementation of the AFM [26].

To better understand the Knowledge Components (KCs) categorized through clustering, the most frequent name within each cluster was selected as the representative name for that cluster, alongside the descriptions included in that cluster. As illustrated in Figure 4, these names typically consist of 2-3 words, such as **Trigonometric Relationships** or **Moment Calculation**. In most clusters, approximately one-third of the items shared the same name, which was then chosen as the mode and designated as the cluster's representative name.

3.4 Knowledge Tracing Baselines Using KCs

Using the extracted knowledge components, we used code from prior work [11] to conduct KT with IRT, PFA, DAS3H, SAKT, and DKT. In our implementation of PFA, we also considered incorporating a student-specific intercept—an enhancement reported to boost performance in recent work [6]—but observed degraded results across all configurations; consequently, we omitted student intercept encoding in our final PFA setup.

In the experiments of the previous section, the performance of the OpenAI embedding model was superior to that of the T5-XXL model, so we evaluated the KCs generated using the OpenAI *text-embedding-3-large* model. Among these, IRT is an algorithm that does not use KCs, while PFA and DAS3H use a Q-matrix. SAKT and DKT operate in environments with disjoint KCs, so we used joint skill assignments. Joint skill considers the combination of KCs assigned to each question as a single KC [44]. For example, a problem with KC 1 and KC 2 is treated as having a distinct KC different from a problem with only KC 1.

To compare the impact of the KCs generated by our algorithm, we prepared four types of baselines. The **Random** setting assigns all KCs randomly as a baseline. The **Human** setting uses the highest-performing human-generated KCs tagged in each dataset from CMU Datashop.

In addition, we also measured the zero-shot KT performance on completely unseen items. To achieve this, we created a train-test split where no items overlap between the train and test sets, but the KCs of the items in the test set appear at least once in the train set. Then, we applied logistic regression to the KCs-aware features to determine the zero-shot performance across each domain. By splitting the dataset in this manner, we ensured that the model's ability to generalize to new items was rigorously tested. This approach allowed us to evaluate the robustness and adaptability of our LLM-generated KCs in handling novel content.

3.5 Detailed Experiment Settings

3.5.1 LMM Inference

Figure 5 shows the prompt used as the system role. The random seed for the API was set to 42, and the problem was provided in the user role. The problem was appended with the postfix 'Content:\n\n'. Images were uploaded and

---

Prompt

Extract the knowledge components required to solve this question. Each knowledge component has two fields:

- Name: 2 to 4 words

- Description: 1 sentence

Output is in JSON format, like:

```
{
  "knowledge_components": [
    {
      "name": "knowledge component 1",
      "description": "understand
      how to apply kc 1"
    }
  ]
}
```

Figure 5: GPT-4o prompt used for knowledge components extraction

---

encoded in base64, and the prompt was generated to place the images between the text, preserving their positions as closely as possible to their actual locations in the problem.

3.5.2 Zero-Shot Knowledge Tracing

We implemented zero-shot knowledge tracing using the same codebase[11] that was used for KT implementation[11]. The logistic regression setup was configured to experiment with various combinations of features, specifically utilizing the s, sc, tc, tw, w, and a tags. These options allow us to use features that record which KCs are present, how many times each KC appears in the user's history, the total number of problems the user has solved, the total number of problems the user has correctly solved, and how many attempts were made for each problem within a time window. When selecting these options, we understood the code and chose all relevant options, but did not compare with other options due to the absence of a validation set.

3.6 Quality of the Generated Knowledge Components

3.6.1 AFM Analysis

For AFM inference, we used the Python implementation compatible with the CMU DataShop format[12]. The mapped KCs were post-processed in the same scheme as when transactions are exported with the 'By Transaction' option from CMU DataShop, and the inference was performed using this code. We were unable to conduct AFM analysis on the Psychology and Biology datasets because the memory usage of the code increases proportionally with the transaction length. The required memory exceeded 40GB, which could not be handled by the computational resources used in this study, specifically a MacBook M3 Pro with 36GB of

---

[11]https://github.com/theophilegervet/learner-performance-prediction
[12]https://github.com/cmaclell/pyAFM

Table 1: Examples of assigned Knowledge Components and the descriptions

(a) oli_statics

Generated Name	Description	KC Name
Item ID: 1273		
Reading comprehension	Understand the text of the question and the options provided.	identifying correct option
Multiple choice format	Recognize the structure of a multiple choice question and how to select an answer.	understanding question format
Decision making	Decide between the given options based on the question's requirements.	identifying correct option
Item ID: 1097		
Summation of Forces	Understand that $\Sigma F\ y = 0$ denotes the summation of all forces in the y-direction equaling zero.	Force Equilibrium
Force Equilibrium	Recognize that the condition $\Sigma F\ y = 0$ implies a state of force equilibrium in the vertical direction.	Force Equilibrium
Multiple Choice Questions	Know how to interpret and answer multiple-choice questions.	multiple choice format
Selecting Correct Answer	Identify the correct option based on given conditions and context.	identifying correct option

(b) oli_psychology

Generated Name	Description	KC Name
Item ID: 2066		
Human Eye Anatomy	Understanding the different parts of the human eye and their functions.	Young-Helmholtz theory
Iris Location	The iris is positioned between the cornea and lens, controlling the amount of light that enters the eye.	Young-Helmholtz theory
Item ID: 709		
Reasonable Mind Concept	Understanding that the decision involves logical planning and time management.	True or False Questions
Wise Mind Concept	Recognizing the balance between emotional and reasonable mindsets, though not applicable here.	Identifying emotions
Emotional Mind Concept	Understanding that decisions driven purely by emotions are not being considered in this scenario.	Identifying emotions
Time Management Skills	The ability to plan and allocate time effectively for various activities including work, study, and social events.	True or False Questions

shared memory. We used 3-fold cross-validation, adopting the default hyperparameters from the original AFM implementation [26].

4. RESULTS

Figure 4 provides an example of generated KCs descriptions classified into the same cluster, represented as *Trigonometric Relationships*. Figure 6 shows the performance change of AFM with varying cluster numbers. When item information is masked, performance relies solely on the general ability of each student and KCs information, making it highly dependent on KCs quality. While overall and student-masked performance improved with more clusters, item-masked performance deteriorated, indicating that more clusters do not necessarily mean better KCs and suggesting room for improvement in verifying KCs consistency.

Table 3 compares the AFM performance of KCs selected using OpenAI's embedding API, Sentence-T5-XXL embeddings, and human experts in the *statics*, *french*, and *computing* domains. The results using OpenAI's embeddings consistently outperformed Sentence-T5-XXL. Given that previous research has shown the superior performance of OpenAI's embedding models [3], it can be concluded that as em-

bedding models improve, the performance of KCs is likely to improve as well. However, both methods still showed comparable or better overall performance than human-created KCs, while item-blocked cross-validation performance was worse.

In the context of automatically generated KCs, the item-blocked performance of the AFM tends to be somewhat lower compared to that observed with human-defined KCs. We attribute this phenomenon to the AFM's reliance on the Opportunity Count feature. As the number of tags increases, the opportunity count values input for each prediction converge toward zero, leading to a sparsity of information that can adversely affect model performance.

As demonstrated in Table 2, all three subjects experienced a substantial increase in the number of generated KCs compared to the original sets. Specifically, Figure 6 provides compelling evidence that item-blocked performance diminishes as the number of KCs escalates. This trend is further supported by the significant disparity in AFM performance observed in the oli_french dataset, where the original human-created KCs numbered only seven. In the computing dataset, the KC count increased from 41 to 118, and in stat-
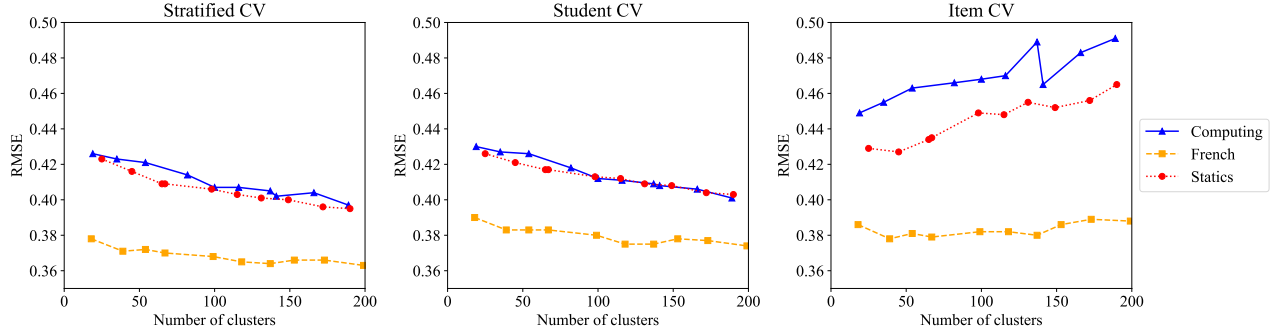
Figure 6: AFM performance change based on the number of clusters.

Table 2: Summary statistics of generated KCs for different OLI courses.

Course	# KCs	Avg. KCs per item	% Multi-KC	# KCs in gold
oli_computing	118	3.14	91%	41
oli_statics	187	3.01	94%	81
oli_french	196	2.91	94%	7
oli_biology	197	2.1	71%	275
oli_psychology	187	2.08	69%	226

Table 3: AFM scores. RMSE columns are the full cross-validation score, and - Student and - Item columns are the performance when the corresponding feature is blocked. *openai* and *T5-XXL* are our generated KCs, while the others were created by humans.

Model	RMSE	- Student	- Item
OLI_statics			
openai	0.395	**0.403**	0.465
T5-XXL	0.395	0.404	0.478
F2011	**0.394**	**0.403**	**0.407**
OLI_french			
openai	0.363	0.374	0.388
T5-XXL	0.376	0.385	0.409
Bonnie	**0.345**	**0.354**	**0.346**
Level4	0.354	0.358	0.355
OLI_computing			
openai	**0.397**	**0.401**	0.491
T5-XXL	0.398	0.402	0.502
poc_1_13	0.416	0.422	**0.432**
poc_1_10	0.428	0.433	0.435

ics from 81 to 187—approximately doubling in both cases. In contrast, the oli_french dataset saw an increase from 7 to 196 KCs, a 28-fold amplification, which likely intensified the observed effect on AFM performance.

Thus, while the AFM is capable of effectively evaluating performance in multiple-KC systems, we infer that significant discrepancies in the number of KCs can lead to inequitable comparisons between different KC sets. This finding implies that when applying Knowledge Tracing (KT) methodologies, especially in contexts with vastly differing KC counts, it is crucial to consider the potential impact on model performance assessments. Careful examination of these factors is essential to ensure fair and accurate evaluations within educational data mining and learning analytics.

As qualitative aspects of generated KCs, Table 1 presents randomly selected content from the oli_statics and oli_psychology datasets, showing the GPT-4o generated names and descriptions for the KCs tagged to those contents and the KC names classified by clustering. For the Biology and Psychology datasets, the higher specificity of the topics often resulted in all KCs within a single problem being classified under the same tag, explaining the relatively low Multi_KC ratio in Table 2. Figure 4 displays example knowledge components belonging to two clusters in oli_statics dataset.

4.1 Effect on Knowledge Tracing Performance

Table 4 shows the results of the KT experiments. For PFA and DAS3H, which are logistic regression-based KT models that can utilize multiple KCs [33, 4], we find using the KCs generated by our algorithm improves performance compared to the **Random** baseline. Notably, in PFA performance, our KCs outperform those of human experts across three domains. We believe that these advantages stem from the rich information provided by the multiple KCs tags per item.

When comparing model-wise performance, our generated KCs exhibited a pattern similar to that of human expert KCs. In certain datasets and models, using LMM-generated KCs even showed a greater performance increase compared to human KCs. This, along with the previous experiments, supports the conclusion that our generated KCs explain the training data as effectively as human experts.

5. DISCUSSION & LIMITATION

The goal of this work was to evaluate the effectiveness of using LMMs to generate KCs directly from the text, figures, and diagrams of questions. In an empirical evaluation using AFM, we found that the KCs generated by our LMM-based method matched the quality of human-generated knowledge components. Our method worked across five different domains (from computing to psychology to French) and four different knowledge tracing models. Furthermore, in models designed to work with multiple KCs per question, the KCs

Table 4: Knowledge Tracing performance metrics (AUC). The IRT method doesn't use any KC information. Only PFA supports multiple KCs, while the other models concatenate all KCs and treat them as a single, independent KCs.

Knowledge Component Source	KT Model	French	Computing	Statics	Biology	Psychology
None	IRT	0.822	0.809	0.797	0.743	0.781
Random	PFA	0.619	0.604	0.600	0.595	0.590
	DAS3H	0.873	0.816	0.804	0.762	0.793
	SAKT	0.828	0.780	0.812	0.858	0.809
	DKT	0.925	0.817	0.860	0.912	0.822
LLM (Ours)	PFA	**0.787**	**0.723**	**0.751**	0.666	**0.698**
	DAS3H	0.881	0.800	0.836	**0.772**	**0.802**
	SAKT	**0.869**	**0.802**	**0.854**	0.869	0.815
	DKT	0.918	0.835	**0.883**	0.915	**0.828**
Human	PFA	0.752	0.699	0.693	**0.671**	**0.698**
	DAS3H	**0.911**	**0.840**	**0.843**	0.768	0.801
	SAKT	0.850	0.554	**0.854**	**0.874**	**0.817**
	DKT	**0.929**	**0.868**	0.877	**0.918**	**0.828**

generated by our method outperformed human-generated KCs in four of the five datasets. Our method has the potential to immediately improve the quality of intelligent tutoring systems by making it possible to quickly generate high quality KCs for practically any set of questions.

5.1 Towards More Refined Domain Modeling

As reviewed by previous work [35], Modeling the teaching domain with KCs has the potential to achieve greater granularity and richer informational content by assigning multiple KCs to a single item, modeling hierarchical relationships among KCs, or capturing prerequisite structures. However, in practice, it has often been the case that KCs are directly derived from curricula classifications, leading to a one-to-one mapping between problems and KCs [42]. Consequently, Transformer-based KT models, such as SAINT and SAKT, have been designed to utilize this straightforward mapping [5, 32]. However, given that real-world problems typically require the interconnection of multiple KCs, this approach deviates significantly from the intended role of KCs as *"units of cognitive function"* within the original Knowledge-Learning-Instruction framework [19].

Moreover, recent studies have increasingly diverged from the original definition of KCs, treating them as mere metadata to be leveraged for enhancing KT performance. There have been instances where KCs have been used interchangeably with terms like *knowledge concepts* [2] or generalized into labels such as *knowledge, tag,* or *skill* [42]. This shift risks undermining the connection between KCs and learning science fields, such as recommendation systems that rely on KCs to provide meaningful educational insights. Therefore, it is crucial to maintain the integrity of KCs, ensuring they are not reduced to simple metadata for performance enhancement purposes. We note that in this work we use "Knowledge Components" as our primary term and subsume related notions (e.g., concept, skill, tag) under this unified definition.

As large language models continue to advance, enabling the successful execution of more complex cognitive tasks, there is a growing opportunity to construct more sophisticated KC frameworks. These frameworks could consider hierarchical and prerequisite relationships with significantly reduced overhead, offering more robust and nuanced models for educational contexts. Moving beyond the AFM, there is a pressing need for enhanced methodologies that can leverage highly detailed KCs and evaluate their quality. Revisiting discussions on Cognitive Diagnosis Models may also help us remain focused on the core issues at hand [10].

5.2 Linking to Zero-Shot Knowledge Tracing

A key direction for future research, as proposed by this study, is the development of Zero-Shot KT techniques mediated by KCs. Currently, KT models are predominantly optimized for interaction data, rendering assessment infeasible without prior records. This limitation is particularly problematic for ITS, where KT models a student's knowledge state based on their problem-solving history to predict future performance [1]. Despite the advances brought about by transformer architectures, these models still rely heavily on the statistical properties derived from problem-solving records. This approach contrasts sharply with human educators, who can intuitively assess knowledge and identify deficiencies without the need for extensive data histories.

The primary challenges associated with current KT methods include:

1. The inability to manage educational content or students without prior records, leading to cold-start issues for both users and items.

2. Biases stem from an over-reliance on statistical data, which can be influenced by the difficulty level of the content or peer interactions.

3. In the case of Deep Neural Network models, operations that are theoretically expected are not always guaranteed. For example, even if a learner answers more questions correctly than before, the learner's KT prediction value may be lower than it was before answering those questions [17].

While human educators also face challenges with user-cold starts, they can manage biases more effectively by assessing knowledge within learning materials and identifying the

essential problem-solving skills. However, the subjectivity inherent in human assessment remains a significant issue, particularly in large-scale educational systems that struggle to rely solely on human evaluation. In this context, LLMs, with their capacity to quickly comprehend content and analyze vast amounts of data, offer a promising alternative.

The most significant synergy between automatically generated KCs and Zero-Shot KT lies in the potential to enhance KC frameworks without the constant need for human expert evaluation. By leveraging performance comparisons across existing KT benchmarks, much like the automated KC evaluation platforms provided by Datashop based on the AFM [18], researchers can accelerate the research cycle, eliminating human bottlenecks and fostering rapid advancements in educational technology.

5.3 Limitations

Our study has several limitations. First, our data preprocessing led to significant losses—extraction from outdated HTML (including deprecated Flash content), potential audio-to-text conversion errors, and the exclusion of uncertain mappings reduced transaction data by more than half. Second, current KT models (SAKT and DKT) that rely on joint skill modeling are not sufficiently advanced to fully assess our detailed multi-KC approach. Finally, due to the visual nature of some problems, we relied on images rather than text, which prevented direct comparisons between Large Language Models and Large Multimodal Models.

5.3.1 Losses in Data Preprocessing

Refining the content data proved challenging, resulting in substantial losses. Extracting images from HTML files including deprecated Flash elements and converting audio to text introduced errors. The most substantial data loss occurred during the mapping of content data to transaction data. Due to incomplete metadata and the absence of content for some problems, we excluded any content with uncertain mapping to ensure data accuracy. This process reduced the original transaction data by over half.

5.3.2 KT Model Limitations

SAKT and DKT rely on joint skill modeling, which does not fully capture the nuances of our multi-KC approach. Additionally, AFM—being an older model—utilizes relatively simple feature engineering to process time-series data, limiting its ability to evaluate advanced KC details. Our primary focus was on developing detailed KCs, leaving further refinement of KT methodologies for future work. Moreover, it would be desirable for future work to support additional KT frameworks—such as the multi-method pyKT library [24] and the LKT framework [34]—so that our extracted KCs can be evaluated across a broader and more diverse set of models.

5.3.3 Access to Raw Content Data

While we have made all generated tags and refined data publicly available, the parsed raw data remains accessible only through CMU DataShop due to their policy. Because these contents are actually used in educational settings, they cannot be made publicly available without restrictions. However, obtaining access is straightforward and promptly processed, and we have provided reproducible preprocessing code to facilitate this. We are committed to ensuring the usability of this benchmark by providing all necessary support for access.

5.3.4 Upper Bound on the Number of Clusters

As shown in Table 6, AFM's overall RMSE continues to decrease even up to 200 clusters. In theory, a reversal would be expected — an increase in RMSE due to overfitting — when the cluster count becomes sufficiently large, but our experiments were restricted to at most 200 clusters. Moreover, we evaluated only three relatively small datasets (Statics, Computing, and French), which further limited our ability to explore higher cluster counts. These restrictions stem from the computational burden that grows with both the size of the dataset and the granularity of the cluster under finite resources. We anticipate that the development of more efficient algorithms for selecting the optimal number of KCs would enable exploration beyond this bound and could yield additional performance gains.

6. CONCLUSION

We have presented a novel, zero-shot approach that leverages instruction-tuned large multimodal models (LMMs) to automatically extract knowledge components (KCs) from educational multimedia content. Unlike traditional methods—which rely on human-generated labels or purely statistical techniques—our approach directly parses text, images, and audio to generate detailed KCs. Experimental evaluations across five domains and multiple knowledge tracing (KT) models (including IRT, PFA, DAS3H, SAKT, and DKT) demonstrate that the LMM-generated KCs not only match but often exceed the performance of human-defined KCs, thereby improving the accuracy and interpretability of student performance predictions.

In addition, by releasing refined KT benchmarks enriched with these automatically generated KCs, we provide a valuable resource for the community to further develop advanced KT methodologies. While our findings highlight the promise of automated KC extraction in enhancing intelligent tutoring systems, they also reveal key limitations—such as significant data losses during preprocessing and the constraints of existing KT models in fully capturing the nuances of multi-KC assignments—that must be addressed in future research. Moving forward, integrating more sophisticated domain modeling techniques and exploring zero-shot KT strategies will be crucial for developing more personalized and scalable educational systems.

Overall, our work lays a strong foundation for the next generation of content-aware KT models, bridging the gap between modern AI capabilities and educational practice.

References

[1] G. Abdelrahman, Q. Wang, and B. Nunes. Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023.

[2] F. Ai, Y. Chen, Y. Guo, Y. Zhao, Z. Wang, G. Fu, and G. Wang. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. *International Educational Data Mining Society*, 2019.

[3] H. Cao. Recent advances in text embedding: A comprehensive review of top-performing methods on the mteb benchmark. *arXiv preprint arXiv:2406.01607*, 2024.

[4] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *International Conference on Educational Data Mining (EDM 2019)*, 2019.

[5] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the seventh ACM conference on learning@ scale*, pages 341–344, 2020.

[6] W. Chu and P. I. Pavlik Jr. The predictiveness of pfa is improved by incorporating the learner's correct response time fluctuation. *International Educational Data Mining Society*, 2023.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278, 1994.

[8] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson, and A. Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024.

[9] W. Dai, J. Lin, H. Jin, T. Li, Y.-S. Tsai, D. Gašević, and G. Chen. Can large language models provide feedback to students? a case study on chatgpt. In *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 323–325. IEEE, 2023.

[10] J. De La Torre. The generalized dina model framework. *Psychometrika*, 76:179–199, 2011.

[11] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[12] A. Goslen, Y. J. Kim, J. Rowe, and J. Lester. Llm-based student plan generation for adaptive scaffolding in game-based learning environments. *International Journal of Artificial Intelligence in Education*, pages 1–26, 2024.

[13] S. Grassini. Shaping the future of education: exploring the potential and consequences of ai and chatgpt in educational settings. *Education Sciences*, 13(7):692, 2023.

[14] R. Gubelmann, M. Burkhard, R. V. Ivanova, C. Niklaus, B. Bermeitinger, and S. Handschuh. Exploring the usefulness of open and proprietary llms in argumentative writing support. In *International Conference on Artificial Intelligence in Education*, pages 175–182. Springer, 2024.

[15] J. He-Yueya, G. Poesia, R. Wang, and N. Goodman. Solving math word problems by combining language models with symbolic solvers. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

[16] B. Hu, L. Zheng, J. Zhu, L. Ding, Y. Wang, and X. Gu. Teaching plan generation and evaluation with gpt-4: Unleashing the potential of llm in instructional design. *IEEE Transactions on Learning Technologies*, 2024.

[17] M. Kim, Y. Shim, S. Lee, H. Loh, and J. Park. Behavioral testing of deep neural network knowledge tracing models. *International Educational Data Mining Society*, 2021.

[18] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43:43–56, 2010.

[19] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798, 2012.

[20] U. Lee, J. Bae, D. Kim, S. Lee, J. Park, T. Ahn, G. Lee, D. Stratton, and H. Kim. Language model can do knowledge tracing: Simple but effective method to integrate language model and knowledge tracing task. *arXiv preprint arXiv:2406.02893*, 2024.

[21] A. Lieb and T. Goel. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2024.

[22] F. A. F. Limo, D. R. H. Tiza, M. M. Roque, E. E. Herrera, J. P. M. Murillo, J. J. Huallpa, V. A. A. Flores, A. G. R. Castillo, P. F. P. Peña, C. P. M. Carranza, et al. Personalized tutoring: Chatgpt as a virtual tutor for personalized learning experiences. *Przestrzeń Społeczna (Social Space)*, 23(1):293–312, 2023.

[23] R. Liu and K. R. Koedinger. Going beyond better data prediction to create explanatory models of educational data. *The Handbook of learning analytics*, 1:69–76, 2017.

[24] Z. Liu, Q. Liu, J. Chen, S. Huang, J. Tang, and W. Luo. pykt: A python library to benchmark deep learning based knowledge tracing models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[25] Z. Liu, S. X. Yin, C. Lee, and N. F. Chen. Scaffolding language learning via multi-modal tutoring systems with pedagogical instructions. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 1258–1265. IEEE Computer Society, 2024.

[26] C. MacLellan, R. Liu, and K. Koedinger. Accounting for slipping and other false negatives in logistic models of student learning. In O. Santos, J. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, V. S., and M. Desmarais, editors, *Proceedings of the 8th International Conference on Educational Data Mining*, Madrid, Spain, 2015. Interational Educational Data Mining Society.

[27] H. Moon, Y. Yang, H. Yu, S. Lee, M. Jeong, J. Park, J. Shin, M. Kim, and S. Choi. Evaluating the knowledge dependency of questions. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10512–10526, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[28] E. Mousavinasab, N. Zarifsanaiey, S. R. Niakan Kalhori, M. Rakhshan, L. Keikha, and M. Ghazi Saeedi. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163, 2021.

[29] S. P. Neshaei, R. L. Davis, A. Hazimeh, B. Lazarevski, P. Dillenbourg, and T. Käser. Towards modeling learner performance with large language models. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 759–768, 2024.

[30] J. Ni, G. Hernandez Abrego, N. Constant, J. Ma, K. Hall, D. Cer, and Y. Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[31] H. S. Nwana. Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4):251–277, 1990.

[32] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *International Educational Data Mining Society*, 2019.

[33] P. I. Pavlik, H. Cen, and K. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. In *International Conference on Artificial Intelligence in Education*, 2009.

[34] P. I. Pavlik Jr, L. G. Eglington, et al. Automated search improves logistic knowledge tracing, surpassing deep learning in accuracy and explainability. *Journal of Educational Data Mining*, 15(3):58–86, 2023.

[35] R. Pelánek. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User modeling and user-adapted interaction*, 27:313–350, 2017.

[36] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[37] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[38] J. Salminen, S.-g. Jung, J. Medina, K. Aldous, J. Azem, W. Akhtar, and B. J. Jansen. Using cipherbot: An exploratory analysis of student interaction with an llm-based educational chatbot. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 279–283, 2024.

[39] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011.

[40] J. Shin, H. Yu, H. Moon, A. Madotto, and J. Park. Dialogue summaries as dialogue states (ds2), template-guided summarization for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3824–3846, 2022.

[41] M. Srivastava and N. Goodman. Question generation for adaptive education. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 692–701, Online, Aug. 2021. Association for Computational Linguistics.

[42] B. Sun, Y. Zhu, Y. Xiao, R. Xiao, and Y. Wei. Automatic question tagging with deep neural networks. *IEEE Transactions on Learning Technologies*, 12(1):29–43, 2019.

[43] K. Thaker, P. Brusilovsky, and D. He. Student modeling with automatic knowledge component extraction for adaptive textbooks. In *iTextbooks@ AIED*, pages 95–102, 2019.

[44] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*, 2016.

[45] B. Zhan, T. Guo, X. Li, M. Hou, Q. Liang, B. Gao, W. Luo, and Z. Liu. Knowledge tracing as language processing: A large-scale autoregressive paradigm. In A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, editors, *Artificial Intelligence in Education*, pages 177–191, Cham, 2024. Springer Nature Switzerland.