Jiang, B. et al. (Eds.) (2025). Proceedings of the 33rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Towards Scalable Annotation of Math Word Problems: Knowledge Component Tagging via LLMs and Sentence Embeddings

Chor Seng TAN[a*], Chengwei WEI[a], Jung-Jae KIM[a]
[a]*Institute for Infocomm Research (I^2R), A*STAR, Singapore*
*jjkim@i2r.a-star.edu.sg

Abstract: Mathematical word problems are a crucial component of mathematics education, requiring students to integrate multiple reasoning skills. Annotating these problems with knowledge components (KC) enables better personalized learning, adaptive tutoring, and AI-driven educational assessment. However, manual annotation is time-consuming and inconsistent, limiting the scalability of KC-based learning systems. In this work, we introduce a new labeled dataset of MWPs, derived from ASDiv and GSM8K, with KCs aligned to the Common Core mathematics framework. Using this dataset, we benchmark two different methods to perform automatic KC annotation without any labeled examples, namely LLM KC tagging and SBERT sentence embedding similarity scoring. Our results highlight key strengths and limitations of LLMs in this task, revealing challenges in consistency and reasoning alignment with human labels. We then show that SBERT-based similarity scoring underperforms LLM KC tagging, but can be significantly enhanced by combining the two methods, which addresses their respective limitations. This study provides critical insights into the feasibility of automated KC tagging, laying the foundation for future research in AI-assisted curriculum design and intelligent tutoring systems.

Keywords: Math word problems, automated knowledge component annotation, sentence embedding, large language model, natural language processing

1. Introduction

Mathematical word problems (MWPs) play a central role in mathematics education, requiring students to integrate numerical computation with problem-solving and logical reasoning. Unlike direct arithmetic exercises, MWPs demand an understanding of mathematical concepts, real-world application, and multi-step reasoning. As a result, MWPs are widely used in curriculum design, standardized testing, and adaptive learning platforms to assess students' problem-solving abilities (Verschaffel et al., 2020). However, understanding and solving MWPs effectively depends on mastering a set of underlying knowledge components (KCs) - fine-grained mathematical skills that correspond to cognitive subskills such as addition, multiplication, fraction manipulation, and proportional reasoning.

The concept of KCs originates from cognitive science and educational research, where learning is modeled as the progressive acquisition of fundamental skills (Corbett & Anderson, 1995). By tagging MWPs with appropriate KCs, educators and AI-driven tutoring systems can break down complex problems into their constituent skills, enabling personalized learning and targeted intervention (VanLehn, 2006). This approach is foundational to Intelligent Tutoring Systems (ITSs) - AI-driven platforms designed to provide real-time feedback, adaptive exercises, and automated assessments (Anderson et al., 1995; Pane et al., 2014). ITSs rely on KC-annotated datasets to track student progress, predict learning trajectories, and optimize instructional strategies. However, a key challenge in deploying such systems at scale is the manual effort required to annotate MWPs with KCs, which is often time-consuming and inconsistent.

To mitigate the challenges associated with mapping KCs to assessments, a variety of automated solutions employing machine learning and natural language processing (NLP) have been proposed. These approaches primarily use classification algorithms trained on annotated datasets to predict KC for math word problems. For example, Yaumauchi et al. (2023) used word n-gram features with a vector space model and random forest classifiers, while Shen et al. (2021) used a Task-adaptive Pre-trained BERT model. While these methods reported good results, they rely on substantial, high-quality labeled datasets, which are often scarce due to the labor-intensive nature of manual annotation and the need for domain-specific expertise. Moreover, these models are typically trained on a fixed set of curriculum standards – such as the Common Core – and thus exhibit limited generalizability. When applied to a different curriculum or an alternative set of KCs, their performance may degrade significantly, necessitating retraining or reannotation, which limits their scalability and practical utility.

Recent advances in natural language processing, particularly in sentence embedding models and large language models (LLMs), offer promising pathways to automate the KC annotation process. Sentence-BERT (SBERT) (Reimers et al., 2019) for example allows for semantic comparison between math problems and skill descriptions using dense vector representations without the need for labelled data. Meanwhile, LLMs like the GPT-family of models from OpenAI and Llama models from Meta exhibit strong reasoning capabilities and internalized knowledge of curriculum frameworks, enabling them to infer education concepts directly from problem content. These tools may significantly reduce human effort and enhance the scalability of ITSs. However, their ability to consistently identify and map MWPs to the correct underlying KCs has not been fully explored.

To address this gap, we introduce a new dataset of MWPs from ASDiv (Miao et al., 2020) and GSM8K (Cobbe et al., 2021), annotated with knowledge components aligned to the Common Core mathematics framework. Using this dataset, we benchmark several automated KC tagging approaches. First, we use cosine similarity between SBERT-encoded problem-solution pairs and KC descriptions to score and rank candidate KCs. Second, we evaluate GPT-4o mini, a lightweight yet high-performance LLM from OpenAI (2024), under multiple prompting strategies, to assess its ability to predict the appropriate KC directly, relying solely on its internalized knowledge of curriculum standards without external KC descriptions. Lastly, we introduce a hybrid method that prompts GPT-4o mini to generate KC descriptions from the MWP and then match these generated descriptions to KCs using SBERT similarity scoring.

The main contributions of this work are as follows: 1) we present two datasets based on problems from ASDiv and GSM8K with manual KC annotations from experts based on the Common Core State Standards (CCSS), 2) we benchmark the performance of different NLP models, namely sentence embedding using SBERT and LLM annotation using OpenAI's GPT-4o mini, to perform automated KC annotation without any labelled examples, 3) we show that the two methods of SBERT and LLM annotation, including reasoning-based LLM annotation, show high degree of inconsistency across the two datasets, and thus are unreliable to be used as input to manual validation, and 4) we demonstrate that by combining the two methods we can address the issues of the two methods.

Our study provides a comprehensive evaluation of the strengths and limitations of both embedding-based and generative approaches to KC tagging. Our results show that each method has complementary advantages and combining their outputs yields improved performance. This work contributes to AI-driven education research and the broader goal of automating curriculum-aware reasoning tasks, paving the way for more effective AI-powered learning systems.

2. Background and Related Work

Automating the tagging of Knowledge Components (KCs) for math word problems using natural language processing (NLP) methods has been an ongoing area of research for over a decade. One of the earliest efforts was from Cetintas et al. (2009), who trained a support vector machine (SVM) classifier to differentiate Multiplicative Compare and Equal Group

problems from other types of math problems drawn from a fourth-grade math textbook. Subsequent studies expanded the classification scope to a broader range of KCs. For instance, Karlovcec et al. (2012) demonstrated a text mining approach using a SVM classifier as well as a search engine-based approach with a k-nearest neighbors (KNN) classifier to tag math problems with labels drawn from a pool of 106 KCs. Meanwhile, Pardos et al. (2017) used a skip-gram approach to match math problems with labels from a pool of 198 KCs. While these early works reported promising results, later analysis revealed that the datasets used, which were from the ASSISTments platform, were based on templates and vulnerable to overfitting. Patikorn et al. (2019) pointed out that the models were exploiting spurious textual cues – such as keywords and formatting patterns – rather than learning mathematical concepts, thus limiting generalizable to more diverse datasets.

More recent works leverage language models such as BERT and GPT that are based on the Transformer architecture, which allows for modelling bidirectional dependencies and generating context-sensitive representations of texts. (Vaswani et al., 2017). Tan et al. (2024) showed that training a RoBERTa classifier model to predict KC can achieve strong performance on a diversified math word problem dataset. However, this approach still requires annotated data for training and may not generalize well across different curricula. Moore et al. (2024) employed GPT-4 to generate KCs for multiple-choice questions in Chemistry and E-learning. They developed an ontology to cluster questions that assess similar KCs based on their content. However, their method does not rely on any existing set of KCs but instead uses LLMs to generate a new set of KCs based on the problem texts and compares the outputs to KCs generated by humans. Li et al. (2024) also explored the use of LLMs for KC tagging, particularly focusing on prompt engineering, zero-shot and few-shot settings, and comparing different LLM models. They managed to achieve high accuracies with their LLM annotations. Nevertheless, their dataset was limited to just 12 distinctive KCs, which were also not aligned with any established curriculum. Shan et al. (2024) proposed a prompt-based approach where LLMs are guided to produce annotations in a key-value format, with mathematical terms as keys and their corresponding annotations as values. However, their approach focuses on tagging KCs to mathematical expressions rather than word problems and the respective solutions. Their approach is also not aligned with any established curriculum.

Despite all the advances in automated KC-tagging of math problems, there is still a lack of generalizable methods to annotate MWPs with curriculum-aligned KCs that do not require labelled training data. Our work aims to address this gap by evaluating unsupervised approaches for tagging MWPs with curriculum-aligned KCs.

3. Methodology

3.1 Data Construction

To study the effectiveness of automated KC annotation, we utilize two widely used MWP datasets: ASDiv and GSM8K, which are selected due to their complementary characteristics. ASDiv is a diverse dataset for MWP solving comprising 2,305 elementary-level problems collected from sources like textbooks and educational websites, and includes topics like arithmetic, algebra, and geometry. GSM8K comprises 8,500 arithmetic MWPs from elementary and middle school level that require multi-step reasoning. Topics include basic arithmetic up to rate problems and basic probability.

A critical challenge in mathematical learning systems lies in understanding the fundamental skills required to solve different MWPs. To address this, we engaged external annotators with expertise in grade to middle school level math education to manually annotate each problem in our selected ASDiv and GSM8K datasets with its corresponding knowledge component. The goal was to create a high-quality, human-annotated dataset that can serve as a ground truth for training and evaluating automated KC-tagging models.

The annotators were instructed to use a structured framework based on the Common Core State Standards (CCSS), a set of mathematics benchmarks widely adopted in U.S.

states, aligning each problem with fine-grained skill categories. Each problem was assigned one or more KCs, based on not only the problem text but also the solution as provided in the original dataset. For solutions with multiple expressions, each expression is taken as a step. As both ASDiv and GSM8K comprise only math word problems, we emphasize annotating with KCs that involve solving word problems over KCs that are only related to calculations. For multi-step problems like in GSM8K, annotators are instructed to consider both at the step-level as well as the problem-level before deciding on the appropriate annotation. Annotators were given guidance to select the fewest number of KCs and using the earliest grade KCs that best match the mathematical skill required to solve the problem. For example, a problem that requires two calculation steps with different operations to solve may involve addition/subtraction as a step (2.OA.A.1) and multiplication/division as a step (3.OA.A.3), but we can use a single KC that covers two-step word problems with all four operations (3.OA.D.8) to cover the problem.

To ensure consistency in the annotation among the annotators, we conduct several rounds of calibration where each problem was independently annotated by two annotators until an Inter Annotator Agreement (IAA) of at least 80% is achieved. The resulting dataset provides a rich resource for training and evaluating AI models to automatically infer knowledge components, enabling personalized learning, automated curriculum alignment, and improved feedback mechanisms in ITSs. By benchmarking LLM-based approaches on this dataset, we aim to assess whether AI models can effectively replace or augment human annotators in this critical task.

Table 1. Examples of KC-annotated problems from ASDiv

Problem	Solution	Answer	KC
There are 5 birds in a tree. How many bird legs do you see?	5*2=10	10 (bird legs)	3.OA.A.3
Dave bought a new flat screen TV. The screen was 2 feet wide and 4 feet tall. What is the area of the screen?	2*4=8	8 (square feet)	4.MD.A.3
The sum of three consecutive odd numbers is one hundred twenty-three. What is the smallest of the three numbers ?	x:The first number; x+(x+2)+(x+4) =123	39	6.EE.B.6

Table 2. Examples of KC-annotated problems from GSM8K

Problem	Solution	Step	KC
A chef bought 4 bags of onions. Each bag weighs 50 pounds. A pound of onions cost $1.50. How much did the chef spend?	A bag of onions cost $1.50 x 50 = <<1.5*50>>=75.Therefore, the chef spent $75 x 4 = <<75*4>> = 300 for the four bags of onions. #### 300	1.5*50=75	4.MD.A.2
		75*4=300	4.MD.A.2
John buys 2 packs of index cards for all his students. He has 6 classes and 30 students in each class. How many packs did he buy?	John has 6*30=<<6*30=180>>180 students. So he bought 180*2 = <<180*2=360>>360 packs #### 360	6*30=180	3.OA.D.8
		180*2=360	4.NBT.B.5

The final annotated ASDiv dataset comprises 2,000 problems randomly selected from the original ASDiv problem pool, with 49 unique KCs and 107 unique combinations of KCs, demonstrating the diversity of the dataset. On the other hand, the annotated GSM8K dataset comprises 1,426 randomly selected problems with 4,868 total steps, with the number of steps for the problems ranging from 1 to as high as 8, and an average of 3.36 steps/problem. There were a total of 29 unique KCs and 42 unique KC combinations. Figure 1 shows the breakdown of the grades and domains for the annotations of both datasets. While the grades for both datasets span from kindergarten (K) to Grade 8, the ASDiv has a better distribution of grades across the problems in contrast to GSM8K, which sees a majority of problems concentrated

around Grade 4. In terms of the KC domains, both datasets have majority of the KCs under Operations and Algebraic Thinking (OA) and Measurement and Data (MD). Other common domains include Numbers and Operations in Base Ten (NBT), Ratios and Proportional Distribution (RP), as well as Expressions and Equations (EE). Tables 1 and 2 illustrate some example problems from the two datasets, along with their solutions and the corresponding KC annotations.

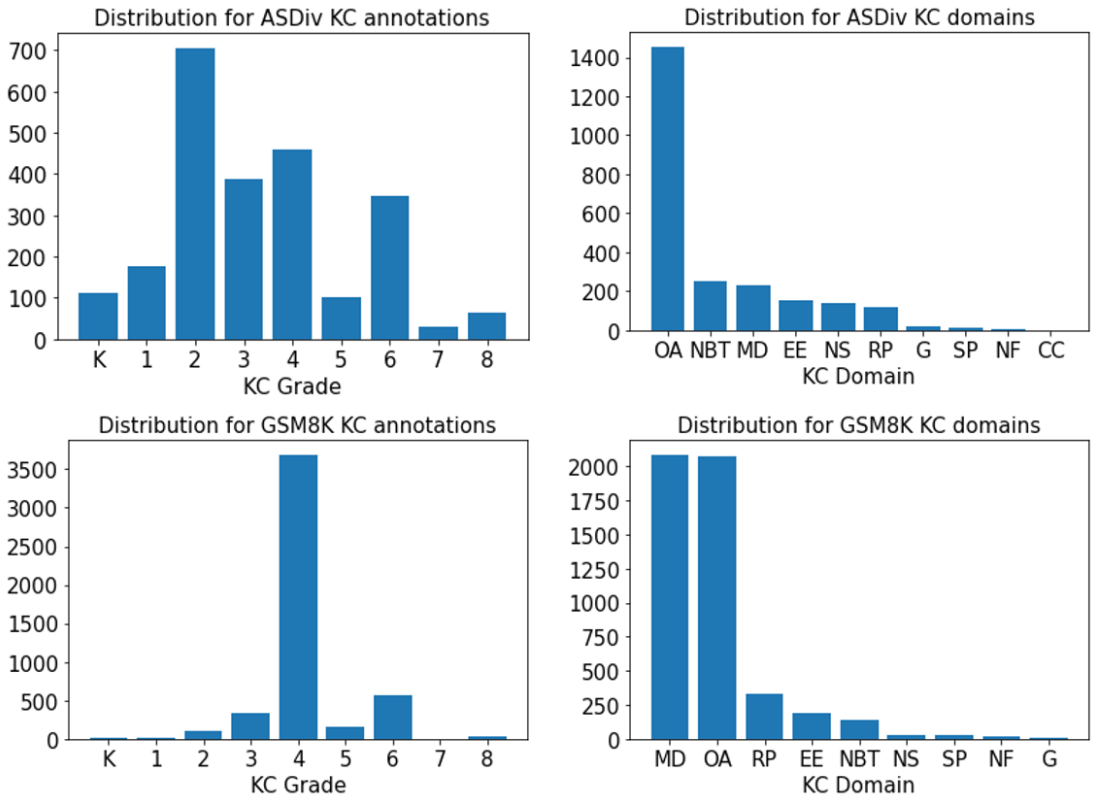

Figure 1. Distribution of grades and domains for ASDiv and GSM8K KC annotations

3.2 Models

3.2.1 – SBERT-based Sentence Embedding and Similarity Ranking

For the first approach, we adopt a retrieval-style strategy using sentence-level embeddings. We use a pre-trained Sentence-BERT (SBERT) model (all-MiniLM-L6-v2) to generate dense vector representations for each problem as well as each candidate KC description from CCSS. For each problem, we concatenate the problem text along with the solution and answer as such: "*Problem/Word problem: <problem text> Solution <solution> Answer: <answer>*". During our experiments with problems from ASDiv, we found that using "Word problem" performed much better than "Problem" as it biases the embedding towards word problem KCs. For GSM8K, we use "Word problem" exclusively for the problem embeddings. We also experimented with using only the problem text without the solution in the problem embedding to compare the performance.

For each problem, we compute the cosine similarity between its embedding and the embeddings of all available KC descriptions. The KC with the highest similarity score is selected as the predicted label. We also evaluate the results using recall@k, such that the top-k ranked KCs can be analyzed to evaluate near-miss performance. We also calculate the mean reciprocal rank (MRR), which captures how close the ground truth is to the top rank. This approach offers a lightweight and transparent baseline for semantic matching between instructional content and curricular standards without requiring any task-specific training.

3.2.2 – LLM-based Labelling with GPT-4o mini

We also leverage on a large language model, GPT-4o mini, to perform automated KC tagging. We prompt the model with the math word problem and the ground truth solution and ask it to select the most appropriate KC from the CCSS. Because there are over 200 KCs in the CCSS just from Grades 1 to 8 alone, we do not provide the model with the descriptions of all the possible KCs. Instead, we rely on the LLM's internal knowledge of the CCSS taxonomy to assign the most appropriate label. This method simulates how an LLM might be used in automated curriculum alignment or tutoring systems, where it must understand both the semantics of the math word problem and the pedagogical scope of various standards without additional input.

We experiment with four prompting strategies:

- **Zero-shot:** A basic prompt that instructs the model to output only the relevant CCSS KC
- **Few-shot**: The prompt includes a few example problems paired with gold KC labels before presenting the test example, the only output is the predicted KC
- **Chain-of-thought (CoT):** The model is prompted to explain its reasoning step-by-step before producing a label KC
- **Few-shot + CoT**: Combines examples with intermediate reasoning steps for a few example problems before prompting the model to reason step-by-step and produce a label KC

These variations allow us to explore the effects of guidance and reasoning on LLM performance in aligning curriculum standards with problem content. We use GPT-4o mini via the OpenAI API, with temperature set at 0 to ensure deterministic output. The outputs produced by GPT-4o mini are parsed using string matching and compared against the ground truth annotations to assess the accuracy. For the chain-of-thought prompts, we also consider the qualitative alignment with human reasoning.

3.2.3 – LLM + SBERT Embedding-based Matching

In addition to the direct SBERT similarity ranking method and the prompting-based LLM classification, we introduce a third hybrid method that leverages the generative capabilities of LLMs together with semantic similarity scoring. This approach is designed to mimic a more human-like reasoning pipeline: first summarizing the mathematical knowledge applied in the problem, then aligning it with an existing curriculum taxonomy.

In this method, we prompt the LLM (GPT-4o mini) with the full math problem text and corresponding solution steps, and instruct it to generate a natural language description of the knowledge components or mathematical skills involved in solving the problem. Once generated, each description is encoded using the SBERT model to produce a vector representation and the cosine similarities between this embedding and those of all the available CCSS KC descriptions are computed and ranked. This hybrid method thus separates the understanding of applied mathematical knowledge, handled by the LLM, from the alignment with standard curriculum concepts, handled by SBERT embeddings, offering a flexible alternative to direct classification or retrieval-based methods.

4. Results

4.1 SBERT sentence embedding results

The results for the SBERT-embedding method on the ASDiv dataset are summarized in Table 3. We see better recall@k across all values for k as well as better MRR when both problem text and the solution are used for the problem embedding. We also observe significant improvements across the board when we use "Word problem" as part of the input for the problem embedding. We observe a similar trend for the GSM8K dataset (Table 4). However,

the overall performance is lower, with a best recall@10 of 51.85% for GSM8K vs. 79.22% for ASDiv, and a MRR of 0.1592 vs. 0.3374. One possible reason for this is that the solution for ASDiv problems only includes mathematic expressions (see Table 1) while for GSM8K there are much longer explanations in natural language along with the mathematical expressions (see Table 2), which dilute the core mathematical concepts applied. As a result, the embedding becomes an averaged representation, and the truly discriminative parts end up being underweighted, adversely affecting the similarity scoring.

Table 3. *SBERT sentence embedding similarity scoring results for ASDiv*

ASDiv	recall@1	recall@3	recall@5	recall@10	MRR
Problem only ("Problem")	10.39%	25.92%	37.91%	58.79%	0.2419
Problem only ("Word problem")	13.19%	33.27%	49.85%	74.08%	0.2964
Problem + solution ("Problem")	16.08%	37.41%	51.20%	68.33%	0.3196
Problem + Solution ("Word problem")	15.53%	40.01%	59.09%	79.22%	0.3374

Table 4. *SBERT sentence embedding similarity scoring results for GSM8K*

GSM8K	recall@1	recall@3	recall@5	recall@10	MRR
Problem only ("Word Problem")	3.04%	8.63%	16.52%	46.24%	0.1395
Problem + Solution ("Word problem")	3.51%	11.50%	24.03%	51.85%	0.1592

4.2 LLM Annotation Results with GPT-4o mini

The results for GPT-4o mini tagging are shown in Tables 5 and 6 for ASDiv and GSM8K problems respectively. For ASDiv, we observe an improvement in accuracy from zero-shot prompting (13.39%) to chain-of-thought (15.68%) as well as few-shot prompting (22.13%). The best performance is achieved by combining few-shot and chain-of-thought prompting, with an accuracy of 27.57%. This compares very favorably with the best recall@1 score for the SBERT-embedding method which achieved 16.08%, and is likely due to the LLM's ability to perform contextual reasoning. One common downside that we observe in LLMs is the tendency to hallucinate, which in this case is usually when it generates KC annotations that do not exist in the CCSS. The rate of hallucination is highest for the zero-shot setting at 6.5%, and is greatly reduced in the few-shot setting at 0.7%. The hallucination rate is moderate at 2.7% and 2.2% for the chain-of-thought and few-shot + chain-of-thought settings respectively.

Table 5. *ASDiv LLM Results with GPT-4o mini*

	Zero-shot	Few-shot	CoT	Few-shot CoT
Accuracy	13.39%	22.13%	15.68%	27.57%
Hallucinations	6.5%	0.7%	2.7%	2.2%

Table 6. *GSM8K LLM Results with GPT-4o mini*

	Zero-shot	Few-shot	CoT	Few-shot CoT
Accuracy	53.51%	54.81%	46.77%	52.59%
Hallucinations	0.2%	0.2%	0.4%	0.2%

In contrast to the SBERT-embedding approach, we observe better results for GSM8K using LLM-tagging compared to ASDiv. The accuracy for few-shot prompting performed the best at 54.81% compared to 53.51% for zero-shot prompting. Unlike for ASDiv, the LLM-

tagging performance is worse with chain-of-thought reasoning, with accuracy of 46.77% for chain-of-thought and 52.59% for few-shot + chain-of-thought. Upon examining the chain-of-thought output provided by the LLM, we observed that the reasoning is usually sound, but the final decision is a similar but incorrect KC. For example, a common error is between 4.OA.A.3 and 4.MD.A.2, which are both related to solving word problems with the four operations, with the former emphasizing multistep problem solving while the latter involving measurement quantities like distances, intervals of time, liquid volumes, and money. However, we note that the LLM accuracy is high for the GSM8K problems compared to ASDiv even for the worse-performing chain-of-thought prompting. The rate of hallucination by the LLM is generally low for GSM8K, ranging from 0.2% to 0.4%.

4.3 LLM KC generation + SBERT sentence embedding

The results for the combined LLM + SBERT-embedding approach are shown in Tables 7 and 8 along with the best results from the other two approaches. The accuracy/recall@1 for ASDiv dataset is much improved from 15.53% using only sentence-embedding similarity scoring and from 27.6% using LLM few-shot + CoT to 39.36% with the hybrid approach. We also see improvements across the board for higher values of k, and the overall MRR achieves 0.5352, meaning that the correct KC is typically within the top 2 ranked KCs on average.

For the GSM8K dataset, the hybrid approach significantly improves the recall@k and MRR over the SBERT-only approach, with recall@1 improving from 3.51% to 16.17%, and MRR improving from 0.1592 to 0.3820. However, the hybrid approach still underperforms the best LLM annotation result, which achieves a recall@1 of 54.81% with few-shot prompting. Upon analyzing the model outputs, we believe this is due to many CCSS KCs having highly similar descriptions, differing only in subtle differences. For example, the description for 2.OA.A.1 and 2.MD.B.5 are very similar in that both are related to solving word problems using addition and subtraction within 100, with the main differences being that 2.OA.A.1 emphasizes that the solution can be up to two steps, while 2.MD.B.5 does not specify the step number but instead focuses on problems involving length with consistent units. When prompted to describe the knowledge components applied in the problem, the LLM may not go into sufficient detail to capture these fine distinctions between similar KCs. As a result, the hybrid approach can retrieve closely related KCs, and perform significantly better than the SBERT-only approach, but struggles to precisely identify the correct one among several similar candidates.

Table 7. *ASDiv results comparison*

ASDiv	recall@1	recall@3	recall@5	recall@10	MRR
LLM Few-shot + CoT	27.6%	-	-	-	-
SBERT Problem + Solution (“Word problem”)	15.53%	40.01%	59.09%	79.22%	0.3374
LLM + SBERT	39.36%	58.74%	72.78%	87.31%	0.5352

Table 8. *GSM8K results comparison*

GSM8K	recall@1	recall@3	recall@5	recall@10	MRR
LLM Few-shot	54.81%	-	-	-	-
SBERT Problem + Solution (“Word problem”)	3.51%	11.50%	24.03%	51.85%	0.1592
LLM + SBERT	16.17%	53.23%	69.78%	82.00%	0.3820

Note that for both datasets, the hybrid approach shows above 50% recall@3 and consistently high performance of recall@k (k=5,10). This is important for manual validation, as the automatic annotations by the hybrid approach provide reliable inputs for manual validation, unlike the two methods of SBERT and LLM annotation. Furthermore, the hybrid

method achieves above 80% recall@10, similar to IAA of human annotators. This would indicate that top-10 results of the hybrid method can be used for manual validation by a single human annotator effectively.

4.4 Discussion

From the results, we observe that while SBERT sentence embeddings can serve as a useful tool for narrowing down candidate KCs for MWPs, their annotation accuracy remains relatively low compared to that of GPT-4o mini, especially for the GSM8K dataset. One likely reason for this performance gap for GSM8K is the difference in solution formats. GSM8K solutions include lengthy, natural language explanations (see Table 2) compared to the more concise, math expressions-based solution format for ASDiv (see Table 1). Preprocessing of the text solutions to extract only the mathematic steps before computing sentence embeddings may yield better results for GSM8K. Additionally, GSM8K emphasizes multistep problems, which might not be fully captured in the embeddings of the problem text and solution. This makes it challenging for similarity-based methods to accurately distinguish between KCs with subtle differences and consistently align problems with the most appropriate KC.

When annotating MWPs using GPT-4o mini, we found that it shows high degree of inconsistency across the two datasets and that the chain-of-thought prompting has opposite effects between the datasets, while few-shot prompting leads to performance improvements for both datasets. The few-shot prompting helps to guide the model towards producing responses that align more closely with the desired output format. Since MWP annotation is inherently subjective and context-dependent, it is crucial to craft clear, targeted system prompts and carefully selected few-shot examples when using LLMs for annotation. These prompt strategies help steer the model towards consistent and curriculum-aligned outputs, thereby improving annotation quality and reliability.

The hybrid LLM+SBERT approach demonstrates promising performance, particularly on the ASDiv dataset. By combining the LLM's ability to abstract and articulate the concepts involved in a MWP with SBERT's semantic similarity matching, this method outperforms both the SBERT-only and LLM-only approaches on ASDiv. This is likely due to ASDiv's well-structured problems and concise solution formats, which make it easier for the LLM to generate focused descriptions and for the similarity scoring to correctly identify the matching KC. In contrast, on GSM8K, the hybrid method underperforms the LLM-only approach. The longer, multi-step reasoning required in GSM8K appears to make the LLM's generated KC descriptions either too verbose or too vague, resulting in less precise matches during similarity scoring.

5. Conclusion

Our experiments demonstrate that automated KC tagging for math word problems by combining SBERT embeddings and LLMs can produce promising results, offering valuable support in reducing manual effort and increasing scalability. Importantly, our findings suggest that further refinement—such as improved prompt design and more carefully curated few-shot examples—has the potential to significantly boost performance. With continued progress, these methods could eventually approach human-level reliability, enabling more efficient and consistent KC annotation at scale.

Acknowledgements

This research is supported by the Ministry of Education, Singapore, under its Science of Learning Grant (award ID: MOE-MOESOL2021-0006). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. The journal of the learning sciences, 4(2), 167-207.

Cetintas, S., Si, L., Xin, Y. P., Zhang, D., & Park, J. Y. (2009). Automatic Text Categorization of Mathematical Word Problems. In FLAIRS.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J. (2021). Training verifiers to solve math word problems. arXiv:2110.14168.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4, 253-278.

Karlovčec, M., Córdova-Sánchez, M., & Pardos, Z. A. (2012). Knowledge component suggestion for untagged content in an intelligent tutoring system. In Proceedings of 11th International Conference on Intelligent Tutoring Systems, pp. 195-200.

Li, H., Xu, T., Tang, J., & Wen, Q. (2024). Automate knowledge concept tagging on math questions with LLMs. arXiv:2403.17281.

Miao, S. Y., Liang, C. C., & Su, K. Y. (2020). A diverse corpus for evaluating and developing English math word problem solvers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 975–984.

Moore, S., Schmucker, R., Mitchell, T., & Stamper, J. (2024). Automated generation and tagging of knowledge components from multiple-choice questions. In Proceedings of the eleventh ACM conference on learning@ scale, pp. 122-133.

OpenAI. (2024). GPT-4o Mini: Advancing cost-efficient intelligence. OpenAI. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Pane, J. F., Griffin, B. A., McCaffrey, D. F., & Karam, R. (2014). Effectiveness of cognitive tutor algebra I at scale. Educational Evaluation and Policy Analysis, 36(2), 127-144.

Pardos, Z. A., & Dadu, A. (2017). Imputing KCs with representations of problem content and context. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, pp. 148-155.

Patikorn, T., Deisadze, D., Grande, L., Yu, Z., & Heffernan, N. (2019). Generalizability of methods for imputing mathematical skills needed to solve problems from texts. In Proceedings of 20th International Conference on Artificial Intelligence in Education (AIED), pp. 396-405.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992.

Shan, R., & Youssef, A. (2024). Using large language models to automate annotation and part-of-math tagging of math equations. In International Conference on Intelligent Computer Mathematics, pp. 3-20.

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., McGrew, S., & Lee, D. (2021). Classifying math knowledge components via task-adaptive pre-trained BERT. In Proceedings of 22nd International Conference on Artificial Intelligence in Education (AIED), pp. 408-419.

Tan, C. S., & Kim, J. J. (2024). Automated Math Word Problem Knowledge Component Labeling and Recommendation. In International Conference in Methodologies and Intelligent Systems for Technology Enhanced Learning, pp. 338-348.

VanLehn, K. (2006). The behavior of tutoring systems. International journal of artificial intelligence in education, 16(3), 227-265.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Verschaffel, L., Schukajlow, S., Star, J., & Van Dooren, W. (2020). Word problems in mathematics education: A survey. ZDM Mathematics Education, 52, 1-16.

Yamauchi, T., Flanagan, B., Nakamoto, R., Dai, Y., Takami, K., & Ogata, H. (2023). Automated labeling of PDF mathematical exercises with word N-grams VSM classification. Smart Learning Environments, 10(1), 51.