One Step at a Time: Progressive Multi-Step Reasoning with LLMs for Automatic Knowledge Tagging

Anonymous ACL submission

Abstract

Knowledge tagging is a fundamental task in intelligent education, associating educational materials with the most pertinent knowledge concepts. However, in practical scenarios, most existing methods have encountered bottlenecks due to the expertise and confusion of knowledge concepts. In this paper, we propose LLM4KTS, which achieves a progressive multi-step reasoning paradigm that fully introduces the reasoning ability of Large language models (LLMs) for knowledge tagging tasks. To build LLM4KTS, we first construct a multi-step reasoning dataset with gradual thinking and reasoning. LLM4KT is then fine-tuned on the dataset to align the LLMs with processive reasoning. Then, we introduce a step-level score preference optimization (SSPO) method to fine-tune the LLM4KT further to improve the effectiveness and quality of reasoning processes. Moreover, we apply a scoring model to expand the inference scaling and guide the decoding process. Extensive experiments verify that LLM4KTS achieves significant improvements in the knowledge tagging performance, outperforming current methods.

1 Introduction

In recent years, the development of intelligent education has led to an astonishing growth in educational data generation. Different from generic data, educational data possesses inherent prior knowledge and native target, that is, it serves for teaching specific knowledge concepts (Chen et al., 2014; Romero and Ventura, 2020; Chen et al., 2022). Consequently, knowledge tagging is an important foundation for effectively aggregating and recommending educational resources. The primary goal of knowledge tagging is to associate educational materials, particularly question resources, with their most pertinent knowledge concepts within a knowledge domain. This task significantly enhances the application of educational materials, thereby supporting more precise and targeted educational interventions (Sun et al., 2018; Nie et al., 2020; Li et al., 2024b).

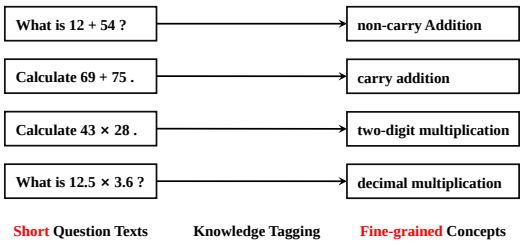

Figure 1: An example of knowledge tagging, where the question text is typically short and the concepts are fine-grained, making knowledge tagging challenging.

Unlike general text tagging, knowledge tagging has several crucial characteristics: Knowledge tagging needs expertise. It demands a profound understanding of the specific domains, including intricate knowledge points and terminology; The knowledge system often comprises detailed and confusable terminology (Zhang et al., 2021; Lee et al., 2023; Liu et al., 2023). For instance, distinguishing between "two-digit addition" and "addition within 10" requires attention; Knowledge tagging tasks frequently require multi-step reasoning processes to uncover underlying knowledge concepts in the question, such as whether addition involves carry. Due to the above characteristics, such knowledge tagging tasks are traditionally performed manually by domain experts. However, manual processes become impractical and unsustainable when facing large-scale datasets. Therefore, developing automated or semi-automated methods for knowledge tagging is critical for advancing educational data analysis in intelligent education.

In the literature, many efforts have been made to conduct automated knowledge tagging. Given that the object of knowledge tagging for questions can be viewed as a text tagging task, some previous works have employed methods such as CRF (Liu et al., 2021), LSTM (Sun et al., 2018),

and BERT-based pre-training models (Zemlyanskiy et al., 2021; Huang et al., 2023). However, there are limitations in these approaches. On one hand, questions typically consist of short texts. Current methods can only extract limited information from the short texts, which is not enough to distinguish accurately. On the other hand, even BERT-based methods lack the necessary reasoning capabilities to recognize subtle differences between fine-grained knowledge concepts as shown in Figure 1 (for example, they cannot differentiate between "non-carry addition" and "carry addition") (Podkorytov et al., 2021). Benefiting from extensive domain knowledge and strong reasoning ability, Large Language Models (LLMs) have achieved impressive results across a variety of tasks including natural language processing, translation, question-answering, and even some professional tasks (Chen et al., 2024; Zhao et al., 2024; Liu et al., 2024). In order to solve the above problems, we adopt the LLM to process knowledge tagging for the questions in a generative way.

In this paper, we draw inspiration from the LLMs and propose a progressive multi-step reasoning paradigm for the question tagging task, especailly knowledge tagging. We make full use of the reasoning abilities of LLMs and effectively use the generative method for knowledge tagging tasks with this paradigm. Firstly, we fine-tune an LLM to obtain our LLM4KT model by introducing the ability of knowledge tagging and aligning the LLM with the given knowledge structures. To make full use of the reasoning abilities of the LLM, we construct a multi-step reasoning dataset for knowledge tagging, which aims to expand the prediction process with gradual thinking and reasoning to obtain a more reasonable prediction result. Then, we introduce a step-level score preference optimization (SSPO) method to further fine-tune the LLM4KT to LLM4KTS. For knowledge tagging tasks with exact results, we conduct a step-level beam search method to obtain the expected score of each step. The higher the expected score, the more likely it is to generate accurate knowledge concepts after the step. In addition, we designed a step-level score preference optimization (SSPO) loss to constrain LLM4KT to improve the generated results at step granularity. Finally, we implement a score-supervised decoding method for score expectations. A scoring model is trained using the dataset with the expected step scores, and it is used to guide the decoding process, choosing the generation with the highest score as the final inference result possible.

We fine-tune LLAMA3.1 to obtain LLM4KT and LLM4KTS following the above paradigm. Experimental results show that our dataset can enhance the performances of LLMs.

The contributions of this paper are:

- We conduct progressive multi-step reasoning with LLMs for knowledge tagging, Experimental results demonstrate the effectiveness and superiority of the LLM4KTS.
- We construct a multi-step reasoning dataset, which divides the knowledge tagging task into a step-by-step thinking and reasoning process. This dataset can better leverage the ability of LLMs. Further, we propose a production pipeline for the dataset with step-scores, and no manual labeling is required in the process.
- We design the step-level score preference optimization(SSPO), which uses the expected step score to more reasonably align the LLM at step granularity.

2 Related Work

2.1 Knowledge Tagging

Knowledge tagging is a fundamental task in educational applications, aiming to establish meaningful associations between knowledge concepts and questions by analyzing stem descriptions or leveraging their solutions. Early studies, such as (Sun et al., 2018) utilized LSTM networks combined with attention mechanisms to encode short-range dependencies in problem statements. Building on this, (Liu et al., 2019) integrated additional features, such as Markov properties, to extract richer concept-related information from exercises. Subsequent work expanded input modalities to include multi-modal data (Yin et al., 2019) and LaTeX-augmented textual data (Huang et al., 2021), significantly improving the ability to capture implicit contextual information. Another line of research introduced knowledge graphs into embedding layers, enhancing semantic understanding in specific domains (Huang et al., 2020). The rise of transformer-based architectures further advanced the field. Zemlyanskiy et al. (2021) leveraged pre-trained BERT models to infer concepts from contextual data, while (Huang et al., 2023) designed a tailored BERT framework to model the interplay between problem statements and solutions, particularly for mathematical problems. More recently,

large language models (LLMs) have emerged as powerful tools for low-resource scenarios. Techniques like chain-of-thought reasoning (COT) and in-context learning (ICL) enable these models to simulate human-like tagging processes, showcasing exceptional adaptability even with minimal annotated data (Li et al., 2024b,c,a). However, current methods still face limitations due to the brevity of question texts, which provide insufficient information for deep differentiation. Even LLM-based approaches struggle with reasoning and fail to distinguish fine-grained knowledge concepts, such as non-carry and carry addition.

2.2 Preference Learning

Recently, preference learning has become a pivotal approach for aligning models with human preferences, offering significant advantages in improving task performance and user satisfaction (Rafailov et al., 2023; Jiang et al., 2024). Early efforts, such as Supervised Fine-Tuning (SFT), aimed to achieve alignment by increasing the likelihood of generating desired outputs (Wei et al., 2022a; Ouyang et al., 2022). However, while SFT enhances the probability of preferred results, it also inadvertently raises the likelihood of undesired outputs, such as hallucinations or logically inconsistent responses. These limitations are particularly pronounced in multi-step reasoning tasks, where errors in intermediate steps can propagate and compromise the final output. To mitigate these issues, Reinforcement Learning from Human Feedback (RLHF) (Zhu et al., 2023; Ouyang et al., 2022) was introduced as a framework for leveraging comparison data to train reward models, which are then used to guide policy optimization. RLHF has demonstrated notable success in generating more reliable outputs by refining models through human-aligned reward signals. However, its complex training pipeline and heavy reliance on the quality of reward models limit its practicality and scalability, particularly for reasoning-intensive tasks requiring fine-grained control (Casper et al., 2023). Direct Preference Optimization (DPO) (Rafailov et al., 2024) offers a more streamlined alternative by directly optimizing models using pairwise preference data, bypassing the need for reinforcement learning. Nonetheless, DPO's performance gains are marginal in domains that require long-chain reasoning, such as mathematical reasoning or tasks involving complex logical steps (Ma et al., 2023). Compared to previous work, our Svpo autonomously annotates step-level preferences through MCTS, and reflects potential reasoning errors through the -values at each step, thereby significantly improving the performance of preference learning on multi-step reasoning tasks.

3 Progressive Multi-Step Reasoning Paradigm

In this section, we present our LM4KTS in detail to further explore the potential of knowledge tagging tasks with the progressive multi-step reasoning paradigm. The problem statement in our paper is as follows: Given the query Q with available text resources of the question, including content, answer, analysis, and so on, and with the prompt to guide our LLM. Our target is to fine-tune an LLM, Θ, with a progressive multi-step reasoning paradigm to generate the result R. Where R consists of step-by-step logical analysis, reasoning, and the selection of the most appropriate knowledge concepts from the given knowledge concept candidates $K = \{k_1, k_2, ...k_m\}$.

To solve the problem mentioned, we conduct LLM4KTS as follows (as shown in Figure 3): First, for questions that need to be tagged, we design a multi-step reasoning prompt to generate the dataset for fine-tuning (Section 3.1). On this basis, we can align the LLM with the expected task. Second, we devise the step-level score preference optimization method to further supervise the LLM4KT to LLM4KTS (Section 3.2). Finally, to further enhance the accuracy of generation, we apply a scoring model to guide the decoding process at step granularity (Section 3.3).

3.1 Dataset Construction and LLM Fine-tune

The most significant difference between LLMs and other prior machine learning models is their reasoning capabilities. By making the large model think step by step, a reasonable context is conducive to helping generate the more accurate result (Wei et al., 2022b; Jin et al., 2024). Further, contributing to the huge size model parameter and the extensive pre-training on diverse and vast datasets, LLMs have demonstrated their strengths in comprehending instructions in natural language and applying learned knowledge to new problems without requiring additional training data specific to these tasks. In our task, to make full use of the reasoning ability and knowledge of LLM, a COT method is adopted to guide LLM to generate the multi-step reasoning for knowledge tagging. We design a multi-step rea-

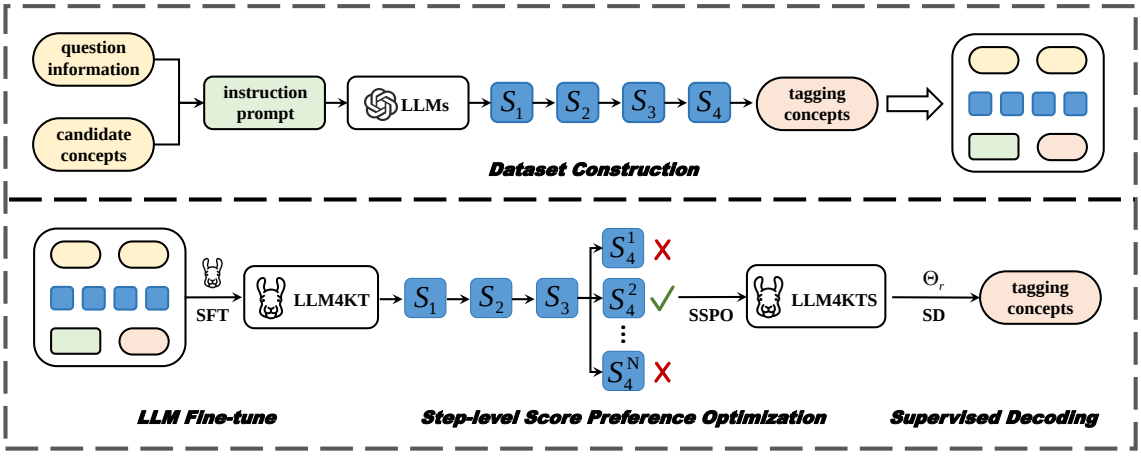

Figure 2: Overview of the LLM4KTS framework.

soning process, which is based on the process of expert analysis and reasoning of questions in practice. A total of four steps are used in this reasoning process, which is: "question analyzing", "preliminary reasoning", "knowledge summarizing" and "name reasoning". Each step contains the step name, the reasoning process, and the reasoning conclusion. The prompts for each step are as follows:

Question Analyzing.

You should comprehensively discover, summarize and analyze all the information in the question, that helps to reason all the detailed knowledge concepts given. Includes but is not limited to the range of values (n digits, multiplication of n, computation within n, etc.), operation methods (addition, subtraction, multiplication, division, parentheses, power squares, etc.), numeric types (integers, fractions, decimals, etc.), important nouns (such as encounter, trace, triangle, etc.) and other hidden conditions.

Preliminary Reasoning.

You should summarize all possible domain of knowledge for the topic, incorporating the information obtained in the question analyzing process.

Knowledge Summarizing.

You should converge the knowledge mentioned in the preliminary reasoning process based on the scope of knowledge associated with the question, and remove the irrelevant knowledge direction.

Name Reasoning.

You should refer to the reasoning of the preceding processes, use the knowledge and nouns mentioned therein, and combine all possible formal knowledge names according to the specification. Finally, you should choose the most reliable knowledge concept among the candidates.

In summary, through such step-by-step reasoning, LLM, like an expert in the field of education, analyzes the knowledge related to the question and deduces the potential knowledge concepts involved. Instead of producing results with direct prompts like "Give a prediction of knowledge tagging directly". Further, LLM generates the selections from the candidate knowledge names with norms. With the multi-step reasoning prompt, the multi-step reasoning dataset is fully automatically generated. Based on the dataset, we can fine-tune a SOTA LLM (e.g., LLAMA3.1) with each sample $\{(Q_1, R_1), (Q_2, R_2), ...\}, R_n = \{S_{n,1}, S_{n,2}, S_{n,3}, S_{n,4}\}$ to LLM4KT. Where the question Q is used as the input, the reasoning result R is used as the target with four steps

$\{S_1, S_2, S_3, S_4\}$. Thus, we align the original LLM to our task to obtain the LLM4KT.

3.2 Step-level Score Preference Optimization

We obtain the LLM4KT with our multi-step reasoning dataset. LLM4KT has the ability to invoke its existing knowledge and generate concept names step by step according to the question. While low-quality generations always exist due to errors in the fine-tuned dataset or direct greedy decoding method. Therefore, we process a preference optimization method to enhance the likelihood of high-quality results in the generations. That is, we optimize the output of the model by further aligning the preferences of the LLM4KT to LLM4KTS with the objective selection preferences (i.e., the accuracy of the results in the knowledge tagging task).

As mentioned above, a reasonable context is conducive to help generate a more accurate result. As with human inductive reasoning about problems, errors in any one premise or step of reasoning can lead to wrong results. Therefore, simply supervising the results or optimizing the whole reasoning process indiscriminately, as we do in fine-tuning, is not enough. After fine-tuning, we then introduce a step-level score preference optimization to align the reasoning by steps.

In the process of preference optimization, the evaluation of data is crucial. Different from subjective tasks, such as free conversation, knowledge tagging is a task with objective results. Therefore, we can accurately evaluate whether a certain reasoning process has reached the correct result. Furthermore, different conclusions may be derived from the same reasoning when the sampling method with certain randomness is used. In this case, we conduct a step-level beam search method to obtain the score of each step in $\{S_1, S_2, S_3, S_4\}$. Obviously, we can directly evaluate the accuracy of a complete and deterministic generation $\{S_1, S_2, S_3, S_4\}$ as $F(S_4) = Score_4$, where is no need to calculate the expected value. Then we merge question Q and parts of generation $\{S_1, S_2, S_3\}$ as the input to only generate the rest tokens in the step of name reasoning repeatedly as shown in Figure 3. It may lead to the $\{S_{4_1}, S_{4_2}, S_{4_3}, S_{4_4}, S_{4_5}, ...\}$. We synthesize all the results as $Score_3 = \sum F(S_{4_i})$, while F evaluates the accuracy of the result. The score is regarded as the expected score of the prefix reasoning $\{S_1, S_2, S_3\}$, and used to indicate the merits of the previous reasoning. Similarly, through such an automated score and expectation calculation process, we get the scores $\{Score_1, Score_2, Score_3, Score_4\}$ for $\{S_1, S_2, S_3, S_4\}$. The scores represent the value of each step in the reasoning process. In particular, the higher the score, the more likely it is that this step (including the preceding steps) will produce superior results. In particular, for different reasoning on the same question, we repeat the above process to obtain the evaluations. For each pair of generation results, we take the sample with a higher step score as a positive sample, thus forming the pairs for preference optimization.

Further, we design the step-level score preference optimization (SSPO) loss to use the dataset with step scores to align the LLM4KT by steps. We use step scores to assign weights to the tokens of different steps, thereby constraining the optimization process. Steps with higher scores should be valued more. For a certain pair, we get the step scores $Score_p = \{Score_1, Score_2, Score_3, Score_4\}$ for positive samples and $Score_n = \{Score_1^{'}, Score_2^{'}, Score_3^{'}, Score_4^{'}\}$ for negative samples. Since the optimization direction of the negative sample is inconsistent with that of the positive sample relative to the reference model in the process of preference optimization. That is, the negative sample should not deviate from the reference model as much as possible, while the changes of the positive sample are encouraged. Therefore, the higher the step score of the positive sample, it means that the change of this step should be focused; Even in a negative sample, there may be a good step with a high score, so the higher the score of a negative sample step, the more we should ignore this step, so as to optimize the bad step and preserve the change of the good step. So we calculate the weights from the scores as:

$$W = Softmax([Score_p, 1 - Score_n]). \quad (1)$$

Finally, we represent the SSPO loss as:

$$\mathcal{L}\left(r_\phi, \mathcal{D}\right) = -\mathcal{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\log \sigma(r_\phi\left(x, w_p, y_w\right) - r_\phi(x, w_n, y_l))]. \quad (2)$$

where $r(x, w, y)$ is the weighted reward function, which measures the difference between the cumulative probability of the sample and the reference model with weight from the step scores as:

$$r(x, w, y) = \beta \log(w \frac{\pi^*(y \mid x)}{\pi_{\text{ref}}\left(y \mid x\right)}) + \beta \log Z(x). \quad (3)$$

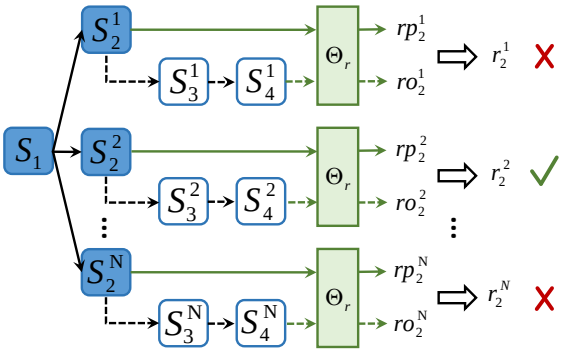

Figure 3: Illustration of supervised decoding, with S_2 as an example.

3.3 Supervised Decoding

We obtain the LLM4KTS with our step-level score preference optimization. In the process of generation, in order to better supervise the process of decoding, we designed a score-supervised decoding process based on process evaluation as shown in Figure 3. We reuse the dataset with step scores in the preference optimization process to fine-tune a reward model Θ_r from LLM. Given the whole or part of the steps, the object of the reward model is to predict the score of the given steps:

$$r = \Theta_r(\{Q, S_1, ...S_n\}). \tag{4}$$

Further, we process a step-by-step decoding with LLM4KTS. In this process, we get N different generation results for the same input Q. We then use the reward model to evaluate the score of the complete result and the latest step:

$$\begin{aligned} ro &= \Theta_r(\{Q, S_1, S_2, S_3, S_4\}), \\ rp &= \Theta_r(\{Q, S_1\}), \end{aligned} \tag{5}$$

where ro is the outcome reward, which measures the correctness of the final result, while rp is the process reward, which provides an estimate of the value of the process. For each sample, we continue to obtain M complete reasoning results. Then we compute the reward r with $r = \alpha * ro + (1 - \alpha) * \sum_{m=1}^{M} rp_m$ as the score of the current result. We select the result with the highest score and continue the reasoning process with the latest step as the input, that is, the input is $\{Q, S_1\}$. We repeat the process to guide the final result. With the supervised decoding, we obtain the result for knowledge tagging with a high evaluation.

Statistics	
# of questions	77476
# of knowledge concepts	12712
Avg knowledge concepts per question	6.0947
Avg length of questions	44.1949
Avg length of prompts	1031.75
Avg length of reasoning process	1121.91

Table 1: The statistics of the MATH dataset.

4 Experiments

4.1 Experimental Datasets

MATH is a dataset from a professional education resource platform, that contains high-quality mathematics questions. Each question contains content, knowledge concepts, and optional analysis and answer content. The details of the data as described in Table 1. Further, we randomly select 5000 questions from the dataset as the testing data, and the remaining data as the training data. In particular, we partition the training data into 90%/10% for fine-tuning part and preference optimization learning part. To conduct the multi-step reasoning data, we use the COT-based prompt to guide the LLM for inference, in which case the average input length of the data including the prompt is 1031.75, and the average inference length of reasoning is 1121.91.

4.2 Experimental Settings

To better illustrate the implementation of our methods, we will introduce the settings in detail. We use the LLAMA3.1-8B as the base model for following the fine-tuning process. In our task, we conduct the candidates with 10 knowledge concepts, that is the $m = 10$. To collect a wide variety of data to conduct the preference pairs in SSPO, we use the different temperatures and topk to generate results. To be specific, (1.0,50),(2.0,50),(0.5,50),(1.0,20) are used. Moreover, in the supervised decoding part, we set the N as 5 and M as 3, the temperature to 1.0, and topk to 10 in the decoding process to provide enough candidates. Then we set alpha as 0.5 to synthesize the different rewards. All experiments are conducted on a server with 6 NVIDIA RTX 3090 GPUs.

4.3 Experimental Results

In this section, we verify the effectiveness of our LLM4KTS by taking GPT4o, GPT4-turbo, LLMALA3.1-8B as the baselines. Each baseline is validated in two ways, one with a progressive multi-step reasoning prompt and one with direct

Method	ACC@3	ACC@5	ACC@10
LLAMA3.1-8B w/o reason	14.74%	12.32%	9.74%
LLAMA3.1-8B	15.56%	12.80%	7.64%
GPT4o w/o reason	39.58%	33.10%	23.88%
GPT4o	54.96%	43.02%	32.26%
GPT4-turbo w/o reason	40.12%	30.76%	22.22%
GPT4-turbo	64.62%	52.14%	36.70%
LLM4KTS	82.31%	72.98%	57.68%

Table 2: Performances comparison on Knowledge Tagging.

Method	ACC@3	ACC@5	ACC@10
LLM4KT w/o reason	51.04%	47.82%	42.04%
LLM4KT	76.55%	66.06%	49.82%
LLM4KT+	75.83%	66.57%	50.15%
LLM4KTS w/o sspo	79.01%	68.58%	52.76%
LLM4KTS w/o sd	81.39%	71.02%	56.81%
LLM4KTS	82.31%	72.98%	57.68%

Table 3: Performances comparison of LLM4KTS and its variants.

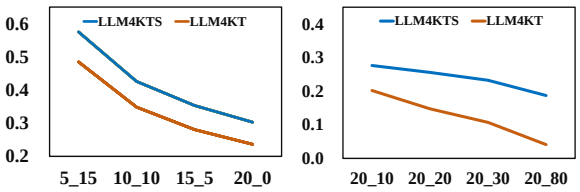

Figure 4: Results of tagging analysis.

generation (w/o reason). In our experiments, our target to verify whether a method is able to pick out the best knowledge concepts from the candidates. Therefore, we design ACC@K to represent the ratio of selecting the correct knowledge concept among K candidates. Specifically, we fine-tune a simple bert model and recalling its top-K predictions with ground truth as the candidates. Compared with existing works in evaluating methods for knowledge tagging, some key observations are as follows: 1) Methods with multi-step reasoning perform better than the methods with direct generation. It shows multi-step reasoning can make full user of the abilities of LLMs and benefit the knowledge tagging tasks. Notablty, our proposed LLM4KTS has the best performances. 2) When the range of candidates is larger, the model encounters severe confusion, leading to worse results, but our proposed method achieves the best results

4.4 Ablation Study

To further validate the effectiveness of the key process in LLM4KTS, we compared LLM4KTS with some variants: LLM4KT w/o reason, LLM4KT, LLM4KT+, LLM4KTS w/o sspo and LLM4KTS w/o sd. LLM4KT w/o reason removes processive multi-step reasoning and is only fine-tuned with direct prompts. LLM4KT is the model which is only fine-tuned with processive multi-step reasoning data. LLM4KT+ is training with all the training data. LLM4KTS w/o sspo processes preference optimization based on LLM4KT without step-level score. In this case, it degenerates into the form of a dpo method. LLM4KTS w/o sd use a simple greedy decoding method to generate the final result. According to the results in Table 3, we obtain the following conclusions: 1) Even after the fine-tune, the method w/o reason performs the worse result than the multi-step reasoning, it shows the multi-step reasoning can enhance the abilities for prediction in LLMs. 2) LLM4KTS w/o sspo achieves better results than the methods with only fine-tune. It demonstrates that preference optimization is conducive to the further improvement of LLM on our task. While LLM4KTS w/o sd has better effect than LLM4KTS w/o sspo, which shows that our SSPO can than provide finer process constraints than direct preference optimization. 3) LLM4KTS achieves the best performance. It shows expanding the inference scaling helps to provide a more reasonable contextual reasoning process and obtain more accurate reasoning results.

4.5 Tagging Analysis

In this paper, we focus on selecting the most appropriate knowledge points from a set of candidates

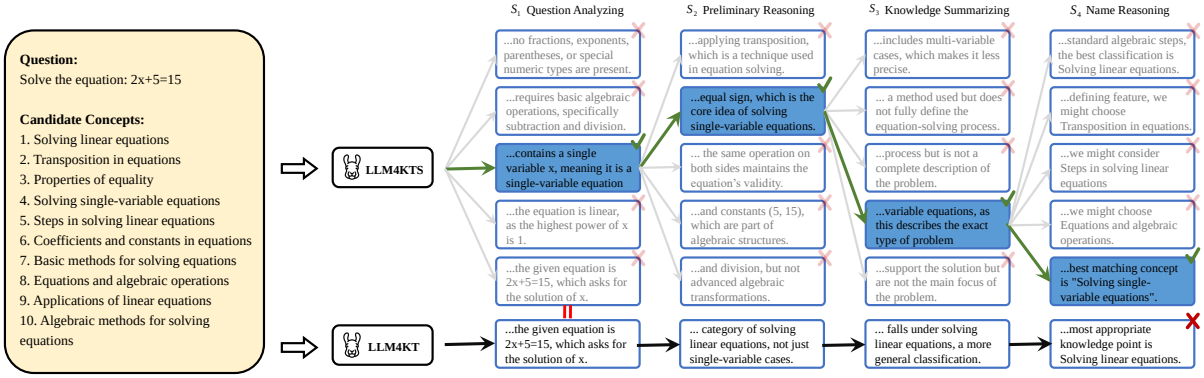

Figure 5: A case study on Knowledge Tagging with LLM4KT and LLM4KTS.

in the context of real-world knowledge annotation. The construction of these candidates is a key factor. In our experiments, we fine-tuned a simple BERT model and used its top-K predictions, alongside the ground truth, as candidates. However, this approach may lack comprehensiveness. We investigate how different candidate construction methods affect the performance and stability of our framework by mixing BERT-selected candidates with randomly chosen ones in varying proportions. BERT-selected candidates are considered "hard" due to their closer resemblance to the ground truth, which increases confusion. We conducted experiments by fixing both the total number of candidates and the number of hard candidates separately. The results, shown in Figure 4, use X_Y notation on the x-axis, where X represents the number of BERT-selected candidates and Y denotes the number of random candidates. Our key findings are: 1) As the number of hard candidates increases, performance declines due to greater confusion. 2) Despite a decrease in the proportion of hard candidates, overall performance still drops with more candidates, likely because the increased selection range outweighs the impact of hard candidates. 3) LLM4KTS consistently outperforms LLM4KT, confirming the effectiveness of our method and the necessity of SSPO.

4.6 Case Study

To illustrate the advantages of our multi-step reasoning framework and process-supervised decoding, we visualize a case study on knowledge tagging with LLM4KT and LLM4KTS. As shown in the Figure 5, while LLM4KT (trained with SFT) can perform step-by-step reasoning, the lack of process supervision may lead to deviations or suboptimal steps, resulting in errors. In contrast, LLM4KTS, optimized with SSPO and equipped with DP, ensures optimal reasoning at each step, enhancing both stability and accuracy. This highlights the effectiveness of our method's design.

5 Conclusion

In this paper, we introduced LLM4KTS, an LLM designed to facilitate processive multi-step reasoning for knowledge tagging. To build LLM4KTS, we defined and conducted a multi-step reasoning dataset, which strengthened the processive reasoning abilities. Besides, we developed a step-level score preference optimization (SSPO) for LLMs and a scoring model was adopted to guide the inferencing. Experiments demonstrated that LLM4KTS significantly outperforms current LLMs such as GPT4. We discuss the broader impacts, limitations, and future work in Limitation.

6 Limitation

In this work, we introduce a step-level preference-based framework for the task of automatic knowledge tagging in educational questions. This framework is not only effective for the specific task but is also easily adaptable and transferable to other annotation scenarios, offering broad applicability. However, the current token processing remains confined to natural language text, limiting support for multimodal data. As a result, predictions for questions containing image-based information may degrade, and the understanding of mathematical formulas often lacks structured depth and hierarchical insight. These limitations highlight the need for further exploration of advanced representation methods to harness better the reasoning and interpretative capabilities of large language models (LLMs).

References

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Jun-Ming Chen, Meng-Chang Chen, and Yeali S Sun. 2014. A tag based learning approach to knowledge acquisition for constructing prior knowledge and enhancing student reading comprehension. *Computers & Education*, 70:256–268.

Xieling Chen, Di Zou, Haoran Xie, Gary Cheng, and Caixia Liu. 2022. Two decades of artificial intelligence in education. *Educational Technology & Society*, 25(1):28–47.

Yuyan Chen, Songzhou Yan, Panjun Liu, and Yanghua Xiao. 2024. Dr. academy: A benchmark for evaluating questioning capability in education for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3138–3167.

Tao Huang, Shengze Hu, Huali Yang, Jing Geng, Sannyuya Liu, Hao Zhang, and Zongkai Yang. 2023. Pqsct: Pseudo-siamese bert for concept tagging with both questions and solutions. *IEEE Transactions on Learning Technologies*, 16(5):831–846.

Tao Huang, Mengyi Liang, Huali Yang, Zhi Li, Tao Yu, and Shengze Hu. 2021. Context-aware knowledge tracing integrated with the exercise representation and association in mathematics. *International Educational Data Mining Society*.

Zhenya Huang, Qi Liu, Weibo Gao, Jinze Wu, Yu Yin, Hao Wang, and Enhong Chen. 2020. Neural mathematical solver with enhanced formula structure. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1729–1732.

Ruili Jiang, Kehai Chen, Xuefeng Bai, Zhixuan He, Juntao Li, Muyun Yang, Tiejun Zhao, Liqiang Nie, and Min Zhang. 2024. A survey on human preference learning for large language models. *arXiv preprint arXiv:2406.11191*.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925*.

Hyun Seung Lee, Seungtaek Choi, Yunsung Lee, Hyeongdon Moon, Shinhyeok Oh, Myeongho Jeong, Hyojun Go, and Christian Wallraven. 2023. Cross encoding as augmentation: Towards effective educational text classification. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2184–2195. Association for Computational Linguistics.

Hang Li, Tianlong Xu, Ethan Chang, and Qingsong Wen. 2024a. Knowledge tagging with large language model based multi-agent system. *arXiv preprint arXiv:2409.08406*.

Hang Li, Tianlong Xu, Jiliang Tang, and Qingsong Wen. 2024b. Automate knowledge concept tagging on math questions with llms. *arXiv preprint arXiv:2403.17281*.

Hang Li, Tianlong Xu, Jiliang Tang, and Qingsong Wen. 2024c. Knowledge tagging system on math questions via llms with flexible demonstration retriever. *arXiv preprint arXiv:2406.13885*.

Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115.

Shuai Liu, Tenghui He, and Jianhua Dai. 2021. A survey of crf algorithm based knowledge extraction of elementary mathematics in chinese. *Mobile Networks and Applications*, 26(5):1891–1903.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024. Are llms capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9215–9235. Association for Computational Linguistics.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Boyu Gao, Weiqi Luo, and Jian Weng. 2023. Enhancing deep knowledge tracing with auxiliary tasks. In *Proceedings of the ACM Web Conference 2023*, pages 4178–4187.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. 2023. Let's reward step by step: Step-level reward model as the navigators for reasoning. *arXiv preprint arXiv:2310.10080*.

Liqiang Nie, Yongqi Li, Fuli Feng, Xuemeng Song, Meng Wang, and Yinglong Wang. 2020. Large-scale question tagging via joint question-topic embedding learning. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–23.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Maksim Podkorytov, Daniel Biś, and Xiuwen Liu. 2021. How can the [mask] know? the sources and limitations of knowledge in bert. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Cristobal Romero and Sebastian Ventura. 2020. Educational data mining and learning analytics: An updated survey. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 10(3):e1355.

Bo Sun, Yunzong Zhu, Yongkang Xiao, Rong Xiao, and Yungang Wei. 2018. Automatic question tagging with deep neural networks. *IEEE Transactions on Learning Technologies*, 12(1):29–43.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yu Yin, Qi Liu, Zhenya Huang, Enhong Chen, Wei Tong, Shijin Wang, and Yu Su. 2019. Quesnet: A unified representation for heterogeneous test questions. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1328–1336.

Yury Zemlyanskiy, Sudeep Gandhe, Ruining He, Bhargav Kanagal, Anirudh Ravula, Juraj Gottweis, Fei Sha, and Ilya Eckstein. 2021. Docent: Learning self-supervised entity representations from large document collections. *arXiv preprint arXiv:2102.13247*.

James Zhang, Casey Wong, Nasser Giacaman, and Andrew Luxton-Reilly. 2021. Automated classification of computing education questions using bloom's taxonomy. In *Proceedings of the 23rd Australasian computing education conference*, pages 58–65.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024. Docmath-eval: Evaluating math reasoning capabilities of llms in understanding financial documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 16103–16120. Association for Computational Linguistics.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, pages 43037–43067. PMLR.