Towards Automated Evaluation of Socratic Tutoring: Introducing IndirectScore for Programming Education

Anonymous ACL submission

Abstract

The Socratic method is an effective pedagogy that uses open-ended questions to foster critical thinking and deeper understanding, but scaling it requires reliable evaluation of question quality, particularly in terms of *indirectness*. In this work, we propose **IndirectScore**, a preliminary automated metric for assessing the indirectness of a Socratic question by leveraging language model surprisal as a proxy for its subtlety. Our approach combines insights from linguistics, NLP, and education to evaluate whether a tutor's question appropriately guides students without being overleading, while explicitly controlling for topical relevance to avoid confounding factors. In an initial evaluation using a newly constructed benchmark of 168 programming dialogues with expert-labeled question quality, **IndirectScore** shows promising alignment of over 71% with human judgments when distinguishing between clearly indirect and direct questions, outperforming traditional NLP metrics such as ROUGE-L and BERTScore. This work represents another step towards scalable evaluation of Socratic questioning, with implications for indirect communication assessment in other interdisciplinary NLP applications. Nonetheless, while these early results suggest potential for building robust AI tutoring systems, we highlight important limitations, such as limited datasets, noise signals, and domain generalisability, and provide directions for future work. [1]

1 Introduction

Socratic questioning plays a pivotal role in education, valued for its effectiveness in distilling knowledge in application-intensive domains such as mathematics, medicine, and computer science (Rodriguez Sandoval et al., 2022; Zou et al., 2011; Le, 2019). This pedagogy relies on scaffolding in which, for a given problem, a tutor guides the student toward the correct solution using structured queries (Quintana et al., 2018).

However, this one-on-one approach, which requires personalized responses from the teacher to every student's answer, is time consuming and resource-intensive (Clark and Egan, 2015). As such, advancement in Language Models (LMs) and other Natural Language Processing (NLP) techniques has sparked interest in leveraging technology to scale the Socratic method (Fakour and Imani, 2025; Wang et al., 2024b), and programming education stands to reap many of those benefits.

Yet unlike factual question generation (QG) tasks, Socratic QG demands *indirect* prompts, questions that do not reveal the answer to the students (Al-Hossami et al., 2023). Shown in Figure 1, the best question allows the student to explore the concepts more freely and induce critical thinking, such as to avoid over-reliance on the tutor (Pit et al., 2024). However, this critical feature currently lacks a robust metric for its evaluation, and whether or not a question is too leading is often relegated to human expert manual annotation (Maurya et al., 2025).

Through a multidisciplinary approach, combining NLP, education, and linguistic theories, we address this evaluation gap and make the following contributions:

- We introduce **IndirectScore**. To our knowledge, it is the first automated metric for quantifying indirectness in Socratic questions. Our metric yields an accuracy greater than 71% against expert labels when comparing the best and worst utterances.
- We release an augmented Socratic dialogue dataset of 56 programming education conversations, derived from Al-Hossami et al. (2023) gold-label corpus. Each dialogue is

[1] All scripts and datasets can be found at: https://anonymous.4open.science/r/IndirectScore-68B4

Student's Task

The CS110Z course director purchased a Disney Vacation Club timeshare that lets you reserve a room at any Disney Resort for one week! It comes with annual "maintenance fee" so that the mouse can keep the property looking good. This year, the maintenance was $623.00, and it accrues each year at a rate of approximately 1.5%.

Write a Python function called 'get_years_until(target_value: float) - int' that takes a target value, and returns the number of years (assuming a fixed interest rate) before the maintenance fee exceeds this value.

Student Code with Bug

```
1 def get_years_until(target_amount):
2     i=623
3     years=0
4     while i < target_amount:
5         years= years+1
6         i= 0.015*i
7     return years
```

Reference Answer

On line 6, the new fee is calculated as 1.5% of the old fee, rather than increasing by 1.5%. As a result, the loop will not terminate. On line 6, replace 'i= 0.015*i' with 'i= i + 0.015*i'

Conversation History

- **Student**: Help! I'm really stuck.
- **Teacher**: Okay, where do you think the problem is?
- **Student**: I don't know.
- **Teacher**: Let's start with your function. What happens on lines 2 and 3?'
- **Student**: I declare some variables for the starting amount and the years.
- **Teacher**: Very good, so let's look at the loop part now. What happens on line 4?
- **Student**: It checks if i is less than the target amount and keeps looping if that's true.'

Best Next Question (Indirect Question)

Teacher: That makes sense, let's look at the calculation part inside the loop. How do we calculate the interest amount?

Medium Next Question

Teacher: Why can "i" never reach the target amount?

Worst Next Question (Direct Question)

Teacher: Are you sure you are calculating the new fee correctly on line 6?

Figure 1: Example conversation with indirect (valid) and direct (invalid) Socratic questions after preprocessing. The Student's Task has been edited to maintain conciseness for display.

manually truncated and supplemented with three follow-up questions, producing dialogue triplets that span the indirectness spectrum (i.e., BEST, MEDIUM, WORST) for benchmarking future automated metrics for indirectness.

- We show that **IndirectScore** significantly differentiates direct from indirect Socratic questions. This metric serves as an initial step toward establishing standardized evaluations of question indirectness.

2 Related Work

2.1 Socratic Question Generation

Socratic dialogues and questions closely resemble sequential information-seeking questions or controlled QG (Reddy et al., 2019; Carlsson et al., 2022), with some educational application such as student assessments (Stasaski and Hearst, 2017; Sarsa et al., 2022). However, the factual nature of these generated questions differs markedly from the reasoning-driven questions essential for effective use of the Socratic method (Paul and Elder, 2008). The most substantial contributions to Socratic-style QG came from Ang et al. (2023), who introduced the *SoQG* dataset, and Shridhar et al. (2022), who focused on mathematical word problems. Both datasets were generated via a semi-automatic process using LLMs. However, the generated questions have not yet been systematically evaluated for their Socratic qualities. More recently, Ashok Kumar and Lan (2024) attempted to automatically generate Socratic questions that were indirect and pedagogically sound. The study utilised a gold-label Socratic code debugging dialogues from Al-Hossami et al. (2023) to produce negative samples, creating an artificially enriched dataset that was then used to finetune an LLM (i.e, Llama 2 (Touvron et al., 2023)) model. The corresponding outputs were then evaluated using ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2019) against larger GPT models, and their smaller model outperformed state-of-the-art models 25 times its size at generating questions that align better with the gold-label dataset. We draw on the produced negative dataset, deemed too direct for student learning, to form part of our evaluation dataset detailed below.

2.2 Socratic Question Evaluation

Acknowledged as a core component of dialectic teaching, indirectness has been investigated in various works, although under different definitions, such as "helping a student" (Tack and Piech, 2022) and "usefulness" (Wang et al., 2024a). Nevertheless, the most concrete definition of indirectness has been offered by Al-Hossami et al. (2023) as part of the four important attributes of a Socratic tutor's question, including:

- *Relevance*: Asking questions that are connected to the topic being discussed
- *Non-Repetition*: Asking questions that have not previously been answered by the student
- *Indirectness*: Asking questions that do not reveal the answers to the student

- *Non-Prematurity*: Asking the right questions at the right time based on the progress of the conversation.

Although Socratic tutoring is much less covered within the broader field of Natural Language Processing, several aspects of its functionality have been extensively studied with respect to various other applications (Favero et al., 2024). In particular, automatic evaluation metrics for assessing a question's relevance and repetition in dialogues are well established, ranging from basic approaches such as n-gram overlap and embedding distance to more advanced techniques such as "three-way attentive pooling". (Kundu et al., 2020; Mikolov et al., 2013; Salkar et al., 2022).

However, previous research has largely overlooked the development of automated metrics to assess indirectness, often defaulting to traditional NLP metrics such as semantic similarity and dialogue ranking (Tack et al., 2023).

2.3 Measuring Indirect Responses

Indirect speech acts have been studied at least since Searle (1975)'s foundational work on speech-act theory. Speakers often flout cooperative maxims to generate implicatures, thereby communicating additional propositions beyond the literal content of an utterance (Grice, 1975). For example, given the question "Do you want to eat?", we have:

- Direct response: Conveys exactly the proposition asked, without additional information (e.g., "Yes" or "No").

- Indirect response: Deviates from the maxim of relation or quantity to add further propositional content, relying on implicature (e.g., "I'm on a diet").

Recent empirical work shows that indirect responses are associated with greater uncertainty and lower coherence than direct replies, making them less predictable in a conversational context (Boux et al., 2022). This unpredictability can be formalized as the divergence between the distribution of responses given a question and the baseline distribution of all possible responses, for example via KL-divergence measures (Zhang and Danescu-Niculescu-Mizil, 2020). Furthermore, this concept has been investigated in the context of conversational uptake, where point-wise Jensen-Shannon Divergence (pJSD) was used to measure the predictability of student responses to a teacher's prompts (Demszky et al., 2021). To quantify this predictability, the study leveraged the cross-entropy loss of an LM as an estimator of the pJSD. Their results showed that the estimator scores were strongly correlated with instructional quality in all analyzed datasets, suggesting that the method is not only reliable across different learning environments but also holds promise as an automated method for assessing the effectiveness of teaching strategies. This success, on top of the theoretical framework on indirect speech mentioned above, forms the motivation for our current approach.

3 Methods

3.1 Constructing the Evaluation Datasets

We employ the *v2 sigcse* benchmark dataset introduced by Al-Hossami et al. (2023) to evaluate the indirectness of Socratic questioning in AI tutoring contexts. This dataset, developed with expert educators, comprises $N = 56$ unique annotated student–tutor conversations, each simulating realistic interactions in introductory programming education.

As Figure 1 illustrates, each conversation $i \in \{1, \ldots, N\}$ is represented by a tuple of four elements:

$$\mathcal{D}_i = (Q_i,\, C_i,\, A_i,\, H_i)$$

where:

- Q_i is the student's task,

- C_i is the student's code with a bug,

- A_i is the reference answer,

- H_i is the full student–tutor conversation history that ends with the student correctly solving the bug.

Each conversation history H_i consists of an ordered sequence of alternating student and tutor turns, indexed from 1 to T_i, where T_i is the total number of turns. Each turn is a tuple $\tau_i^{(t)} = (\text{role}_i^{(t)}, U_i^{(t)})$, where $\text{role}_i^{(t)}$ indicates the role of the speaker, and $U_i^{(t)}$ is set of utterances or responses produced at turn t.

$$\text{role}_i^{(t)} = \begin{cases} \texttt{Student} & \text{if } t \text{ is odd,} \\ \texttt{Tutor} & \text{if } t \text{ is even,} \end{cases} \quad \text{for } t = 1, \ldots, T_i.$$

For each turn $\tau_i^{(t)}$ with $\text{role}_i^{(t)} = \texttt{Tutor}$, the dataset provides up to 5 alternatives:

$$U_i^{(t)} = \left\{ U_{i,1}^{(t)}, \ldots, U_{i,k}^{(t)} \right\}, \quad k \leq 5$$

These alternatives are ordered by the perceived level of indirectness, based on professional annotators' judgments:

$$\text{Indirectness}\left(U_{i,1}^{(t)}\right) > \cdots > \text{Indirectness}\left(U_{i,k}^{(t)}\right)$$

Here, $U_{i,1}^{(t)}$ is the primary response, considered the most Socratic or exploratory, while $U_{i,k}^{(t)}$ offers the most explicit guidance, reflecting the expert evaluations of how much each utterance prompts student thinking versus delivering hints.

Preprocessing Conversation History. To prepare the dataset for evaluation, we construct a truncated and standardized conversation history for each conversation tuple. The truncation is done to remove the portion of the original conversation where the student has deduced the correct reference solution. This pivotal point is removed to simulate a realistic scenario where the student is still exploring the problem. Let $H_i = \left[\tau_i^{(1)}, \tau_i^{(2)}, \ldots, \tau_i^{(T_i)}\right]$ denote the original history. We define a truncated history as:

$$H_i^* = \left[\tau_i^{(1)}, \tau_i^{(2)}, \ldots, \tau_i^{(t_i^*)}\right]$$
$$\text{where } t_i^* < T_i \text{ and } \text{role}_i^{(t_i^*)} = \texttt{Student}.$$

This ensures that each modified dialogue ends with a student's utterance. To introduce diversity in context depth and to ensure robustness of our evaluation to conversation length, we vary the final truncation index t_i^* randomly across different samples, with the criteria that t_i^* is above the pivotal point where the student has found the correct answer.

Finally, for every tutor's turn before the truncation point, where $t < t_i^*$, we retain only the primary (i.e., most indirect) tutor response:

$$\tau_i^{(t)} = (\texttt{Tutor}, U_{i,1}^{(t)}),$$
$$\forall\, t \text{ such that } \text{role}_i^{(t)} = \texttt{Tutor}.$$

That is, all alternative tutor responses are removed, as in the original dataset, subsequent student replies are grounded only in the tutor's primary utterances.

Next Tutor Question Variants. Shown in Figure 2, following preprocessing, each truncated conversation H_i^* is used to construct three evaluation variants by appending a distinct next question from the tutor, denoted as $NextQ_i$. These variants differ in their indirectness level, labeled as BEST, MEDIUM, and WORST, and are defined as follows:

Let the shared base context for each sample be:

$$Context_i = (Q_i, C_i, H_i^*),$$

We then define three augmented samples per conversation:

$$S_i^{(v)} = \left(Context_i,\ NextQ_i^{(v)}\right),$$
$$v \in \{\text{BEST, MEDIUM, WORST}\}$$

where:

- $NextQ_i^{(\text{BEST})} = U_{i,1}^{(t^*)}$, the primary tutor response from the annotated set at the truncation point,
- $NextQ_i^{(\text{MEDIUM})} = U_{i,k}^{(t^*)}$, the most direct alternative in the same annotated set at the truncation point,
- $NextQ_i^{(\text{WORST})} \in \mathcal{N}_i$, a corresponding low-quality question sampled from the negative set introduced by Ashok Kumar and Lan (2024) mentioned earlier.

As a result, the original $N = 56$ conversations are expanded into $3N = 168$ total samples:

$$\left\{\mathcal{S}_i^{(\text{BEST})}, \mathcal{S}_i^{(\text{MEDIUM})}, \mathcal{S}_i^{(\text{WORST})}\right\}_{i=1}^{N},$$

enabling controlled comparative evaluation across differing levels of indirectness.

Isolating The Effect of NextQ. To further highlight the effect of $NextQ$, we separately modified each sample further to remove all conversation history between the student and the teacher. As such, when $NextQ_i$ is appended into each conversation history, it will be the only utterance. We call this new set `Only NextQ` with:

$$Context_i = (Q_i, C_i)$$

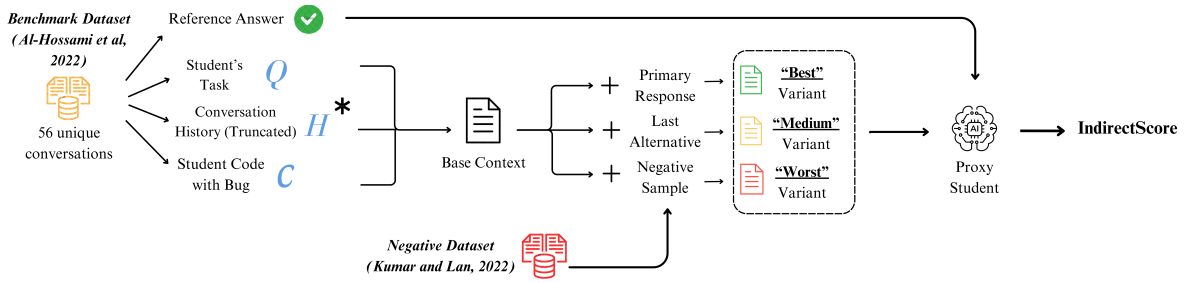

Figure 2: Data Augmentation Process. Each negative question from the dataset by Ashok Kumar and Lan (2024) has a one-to-one match with a corresponding conversation in the gold-label dataset from Al-Hossami et al. (2023). This direct alignment enables seamless substitution of the negative prompt at the truncation point, yielding a low-quality variant for the controlled evaluation.

3.2 Measuring Answer Predictability through IndirectScore

Formal Definition of Indirectness With this, we now provide a more operationalized definition of indirectness. Although the concept is intuitive to human educators, it has lacked a measurable formulation suitable for automated evaluation. Following the modern Socratic pedagogy laid by Maxwell and Maxwell (2014), we define indirectness as the extent to which a Socratic tutor's question withholds the answer in a way that requires the student to perform more intermediate reasoning steps to reach the correct solution. With all other factors held constant, a more indirect question is less likely to elicit the reference answer from the student's immediate response.

Using LLMs as Proxy Students Under this definition, measuring indirectness would require building a distribution of student–tutor responses in programming tutoring dialogues, but the available data is highly scarce. Fortunately, through the recent advancement in LMs trained on large corpus of codes and dialogues, these models can perform as a reliable proxy for students at various levels, especially in single-turn evaluations (Park et al., 2024; Laban et al., 2025). As such, we utilise LMs to approximate that distribution, acting as a simulated student to respond to each variant. We condition these models' behavior through a fixed system prompt, denoted as $SysPrompt$ and shown in the Appendix B.

Thus, to quantify the indirectness of the tutor's next question, we then propose **IndirectScore**, defined as a surprisal of the student model:

$$IndirectScore_i^{(v)} = -\log p\big(A_i \mid SysPrompt, Context_i, NextQ_i^{(v)}\big) \quad (1)$$

This formulation captures the (un)predictability of the reference answer based on the tutor's questions, and as such, a more indirect question should yield a higher **IndirectScore**. To evaluate the performance of the metric, we calculate its accuracy as:

$$\text{Accuracy} = \frac{1}{3N}\sum_{i=1}^{N}\Big[I\big(IndirectScore_i^{\text{BEST}} > IndirectScore_i^{\text{MEDIUM}}\big) + I\big(IndirectScore_i^{\text{MEDIUM}} > IndirectScore_i^{\text{WORST}}\big) + I\big(IndirectScore_i^{\text{BEST}} > IndirectScore_i^{\text{WORST}}\big)\Big] \quad (2)$$

where the indicator function $I(\cdot)$ is defined as:

$$I(\text{condition}) = \begin{cases} 1, & \text{if the condition is true,} \\ 0, & \text{otherwise.} \end{cases}$$

Baselines Because indirectness can easily be mistaken for irrelevance, we first ensured that every $NextQ_i$ selected was demonstrably relevant to its conversational context. This relevance check prevents our metric from rewarding off-topic questions merely because they differ lexically from the answer. Nevertheless, we also include ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2019), applied on $NextQ_i$ and A_i, as baselines for indirectness signals captured by lexical and semantic overlap respectively.

3.3 Experimental Setup

We select two programming-focused language models, Qwen2.5-Coder-14B (Hui et al., 2024) and CodeLlama-13B-Instruct (Rozière et al., 2024), as proxy students, due to their extensive training in code and programming dialogues. Their differing benchmark results also allow us

to explore how model behavior affects **IndirectScore** reliability. All experimental steps were conducted on Nvidia A100 GPUs.

4 Results

4.1 Quantitative Analysis

As shown in Table 1, results from both models outperform traditional NLP metrics, with **IndirectScore** achieving an overall accuracy score of over 60% against expert-labeled data. The performances of both ROUGE-L and BERTScore demonstrate that, by itself, relevance between the question asked and the expected answer simply does not provide a strong enough signal to differentiate indirectness in a tutor's prompt. By leveraging the LMs' output distributions as expected students' responses, our approach mitigates these semantic limitations.

Across variants, **IndirectScore** reveals complex model behaviors. In the full-context setting, Qwen2.5-Coder-14B falters most on *MEDIUM vs. WORST* while CodeLlama struggles with *BEST vs. MEDIUM*. These outcomes likely reflect the subtle distinctions between *MEDIUM* and the more extreme categories, often difficult to distinguish even among human experts (Macina et al., 2023), and remain an open area of research. When limited to the `Only NextQ` as inputs, CodeLlama's accuracy rises almost universally. Qwen, on the other hand, actually shows a decline in performance, likely because its fixed *SysPrompt* leaves it unusually sensitive to the altered input format, yet still exceeds the traditional metrics as baselines. Overall, these shifts suggest that **IndirectScore** detects an informative "surprise" signal embedded in the final question, and that model performance hinges strongly on this last prompt's format.

Amid model-specific fluctuations, a clear trend emerges. **IndirectScore** excels when indirectness contrasts are stark, with *BEST vs. WORST* accuracies exceeding 71% for both models in almost all cases. However, the modest effect sizes in Figure 3 warrant further statistical testing to confirm the robustness of these gains. As such, we further conduct a Wilcoxon Signed-Rank Test to strengthen our findings across all scenarios. This non-parametric approach is appropriate given the paired nature of the test instances, where each pair shares an identical conversational context except for the $NextQ$. Additionally, since the data consists of conversational dialogue, normality cannot be assumed, making the Wilcoxon test a suitable alternative to parametric methods.

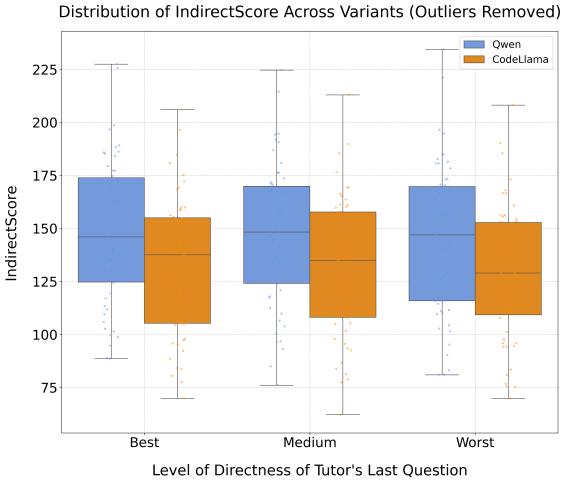

Figure 3: Boxplot for 3 variants across all unique conversations without outliers, constructed from the output scores from the Full Context evaluation dataset.

We compare the **IndirectScore** values of the *BEST* variants against both *MEDIUM* and *WORST* categories. As shown in Table 2, with full context, Qwen2.5-Coder-14B exhibits statistically significant differences, as *BEST* shows higher scores than both *MEDIUM* ($p < 0.05$) and *WORST* ($p < 0.01$). In contrast, while CodeLlama shows a significant difference between *BEST* and *WORST* ($p < 0.01$), it does not yield a significant distinction between *BEST* and *MEDIUM*. Finally, evaluating on the `Only NextQ` condition, it continues to show strong significance when distinguishing between *BEST* and *WORST*, while still showing difficulty in the *MEDIUM* cases.

4.2 Qualitative Analysis

To better understand the strengths and limitations of **IndirectScore**, we also conduct a manual qualitative analysis of question sets, categorizing them into successes and failures based on their scoring patterns. Table 3 provides representative examples from each category. [2]

Analysis of Failures Our examination indicates that these misclassifications (highlighted in red) stem primarily from an answer-type mismatch. Specifically, reference answers are typically descriptive and open-ended, whereas some of the *WORST* questions tend to elicit concise, closed-

[2]Refer to the provided repository for additional examples.

Setup						
Unique Conversations: 56		Total Comparisons: 168				
Metric	**ROUGE-L**	**BERTScore**	**Full Context**		**Only NextQ**	
			Qwen	CodeLlama	Qwen	CodeLlama
Overall Accuracy	30.13%	26.79%	64.29%	65.48%	60.12%	**67.26%**
BEST vs MEDIUM	37.50%	37.50%	**64.29%**	53.57%	48.14%	57.14%
MEDIUM vs WORST	30.36%	26.79%	57.14%	**69.64%**	64.29%	67.86%
BEST vs WORST	22.50%	16.07%	71.43%	73.21%	67.86%	**76.79%**

Table 1: Each row reports the accuracy percentage of instances where IndirectScore assigns a higher score to the expected better variant. Full Context and Only NextQ refers to IndirectScore applied on the sample set with the truncated conversation history and one with entire conversation history removed respectively. For each pairwise category, we remove all other variant comparisons from the calculation.

Wilcoxon Signed-Ranked Test	**Alternative**	**Full Context**		**Only NextQ**	
		W-val	p-val	W-val	p-val
Qwen					
MEDIUM vs BEST	less	575.0	0.035	814.0	0.554
WORST vs BEST	less	393.0	0.001	489.0	0.006
CodeLlama					
MEDIUM vs BEST	less	684.0	0.177	624.0	0.079
WORST vs BEST	less	294.0	0.000	207.0	0.000

Table 2: Wilcoxon signed-rank test results comparing IndirectScore across tutor question variants for each model and context setup. The tests evaluate whether *BEST* yields a higher IndirectScore than other variants under Full and Only NextQ contexts. W-val and p-val represent the test-statistics scores and significance level respectively.

form responses. Addressing this issue might involve rephrasing these prompts to encourage more detailed explanations, as exemplified in the case of "4_28_removing_even_number".

Analysis of Successes via Bloom's Taxonomy The green-highlighted success cases demonstrate instances where **IndirectScore** accurately detects more indirectly phrased questions. We interpret this outcome through Bloom's Taxonomy (Adams, 2015), which links a question's operative verbs to ascending cognitive levels from recall and comprehension to application, analysis, synthesis, and evaluation (Newton et al., 2020).

Shown in Appendix A, an additional mutual-information analysis of token occurrences for all $NextQ$ prompts reveals that *BEST* questions predominantly include open-ended prompts with terms such as "*explain*" and "*let*". These terms stimulate student responses that require analysis, a higher-order cognitive process according to Bloom's Taxonomy. In contrast, tokens frequently appearing in the *WORST* questions, like "*adjust*" and "*modify*" likely prompt basic recall or procedural action. This linguistic distinction reflects a narrower form of student engagement and aligns with the lower levels of Bloom's hierarchy. This alignment of our metric to the well-established framework lends further conceptual support for **IndirectScore**.

5 Discussion

In this study, we introduced **IndirectScore**, a novel metric for evaluating the indirectness of Socratic questions in tutoring dialogues. Grounded in linguistic theory and educational science, and leveraging recent advances in NLP, our initial results demonstrate that **IndirectScore** can distinguish between clearly direct and indirect tutor questions in programming education, outperforming traditional metrics like ROUGE-L and

Filename	Directness Level (Variants)	Next Teacher Question (NextQ)
0_5_fibonacci	BEST	Can you explain what a recursive function is?
	MEDIUM	What happens if you call the fibonacci function with the same argument over and over again?
	WORST	Do you see any problem with the recursive call on line 7?
13_42_limit	BEST	Sure. Can you walk me through what each line does?
	MEDIUM	Sure. It looks like your for-loop is never executed. Do you know why?
	WORST	Are you sure you should be looping over ‘lst2‘ and not ‘lst‘?
17_47_topk	BEST	Ok, no worries. Let’s review your code line by line. Could you please explain it to me?
	MEDIUM	Could you please explain what line 5 in your code does?
	WORST	What if you replace the pop method with the remove method?
4_28_removing _even_number	BEST	Hi, Sure! It looks like your program has a syntax error, do you know what that means?
	MEDIUM	Hi, Sure! Do you know why it is throwing a syntax error on line 3?
	WORST	How can you modify the for loop on line 3 to correct the syntax?

Table 3: Representation of failure and success cases for IndirectScore through Manual Analysis. Misclassified and correctly predictied instances are highlighted in red and green respectively.

BERTScore. Despite the limited dataset, the consistent performances of our metric provides valuable insights into how indirectness can be systematically measured.

Furthermore, while our focus has been on programming education, the work has broader implications for other complex human interactions. In many real-world domains, such as negotiation, counseling, and interviews, indirectness is not just a stylistic choice, but a strategic one that shapes engagement and their outcomes. By quantifying this dimension of communication, **IndirectScore** offers a foundation for evaluating and refining interactional quality across disciplines.

Ultimately, we hope this work fosters further interdisciplinary research into how indirect communication can be systematically evaluated to support more thoughtful, adaptive, and learner-centered AI systems.

6 Conclusion and Future Direction

IndirectScore offers the first automated metric for evaluating indirectness in Socratic tutoring, an essential aspect of the Socratic teaching quality. The metric’s success brings scalable, data-driven refinement of AI tutors closer without relying on costly human annotation.

As future work, we will validate **IndirectScore** on a larger, more diverse dataset. Subsequently, we look to conduct controlled human studies to test whether the questions preferred by **IndirectScore** truly enhance learning outcomes, thereby providing a stronger empirical foundation for our metric’s robustness.

Limitations

Relevance Confound and Metric Robustness

A critical concern when operationalizing indirectness through model-based surprise is the risk of mistaking irrelevant or noisy tutor prompts for pedagogically subtle ones. That is, tutor questions that are off-topic or semantically unrelated to the student's task may still elicit high IndirectScore values due to their unpredictability with respect to the reference answer, thereby inflating their perceived indirectness. As mentioned earlier, we have carefully mitigated this issue through meticulous data curation, manual inspection, and the use of BERTScore and ROUGE-L as baselines. Nonetheless, future work may further formalize this robustness by introducing adversarial irrelevant prompts to stress-test IndirectScore under intentionally misleading conditions or include additional preprocessing to discard irrelvant prompts before proceeding.

Limited Dataset

Due to the scarcity of high quality, labeled educational datasets featuring Socratic dialogue in the programming domain, this study was constrained to a relatively small number of evaluation samples. While additional statistical analyses were conducted to strengthen the validity of the findings, the proposed metric would benefit from more extensive evaluation on larger and more diverse datasets. Expanding the dataset coverage would provide stronger evidence of the metric's generalisability across a wider range of instructional scenarios.

Domain Restriction

This proposed method is inherently restricted to domains where questions have well-defined, objective answers. It is most effective in structured settings such as factual knowledge assessments, coding tasks, and mathematical problem-solving, where correctness is binary or easily verifiable. However, it is not directly applicable to disciplines characterized by open-ended, exploratory, or subjective inquiries, such as philosophy, literary analysis, or qualitative research. In such domains, answer correctness is not absolute but instead depends on interpretative depth, argumentative coherence, or subjective perspectives, rendering log probability metrics less informative.

Model Faithfulness and Prompt Sensitivity

The reliability of the proposed method is inherently dependent on the language model used. Differences in pretraining corpora, model architectures, and fine-tuning objectives can yield divergent output distributions, even among models optimized for similar tasks. For example, state-of-the-art code-focused models such as Codex (Chen et al., 2021) and Deepseek-Coder (Guo et al., 2024), which are not included in this study, differ in their training datasets and optimization strategies, which may influence the probability estimates underlying the **IndirectScore** metric. Further benchmarking across a broader range of models is warranted to evaluate the generalisability and robustness of the proposed approach.

In addition, model behavior is known to be sensitive to prompt design. Although multiple system prompts were evaluated in this study to mitigate this issue, prompt engineering remains an open challenge. Subtle variations in phrasing or instruction style can affect the model's output distribution and, consequently, the computed metric. Future work may explore automated or adaptive prompt optimization techniques to further enhance the stability of the evaluation protocol.

Ethical Considerations

This work entails no discernible ethical or safety concerns, as it utilises publicly available, de-identified datasets drawn from the educational domain. The datasets are used as intended with no foreseeable risks as outcomes.

References

Nancy E Adams. 2015. Bloom's taxonomy of cognitive learning objectives. *Journal of the Medical Library Association JMLA*, 103(3):152–153.

Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan, and Mohsen Dorodchi. 2023. Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 709–726, Toronto, Canada. Association for Computational Linguistics.

Beng Heng Ang, Sujatha Das Gollapalli, and See-Kiong Ng. 2023. Socratic question generation: A novel dataset, models, and evaluation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*,

pages 147–165, Dubrovnik, Croatia. Association for Computational Linguistics.

Nischal Ashok Kumar and Andrew Lan. 2024. Improving socratic question generation using data augmentation and preference optimization. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 108–118, Mexico City, Mexico. Association for Computational Linguistics.

Isabella P. Boux, Konstantina Margiotoudi, Felix R. Dreyer, Rosario Tomasello, and Friedemann Pulvermüller. 2022. Cognitive features of indirect speech acts. *Language Cognition and Neuroscience*, 38(1):40–64.

Fredrik Carlsson, Joey Öhman, Fangyu Liu, Severine Verlinden, Joakim Nivre, and Magnus Sahlgren. 2022. Fine-grained controllable text generation using non-residual prompting. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6837–6857, Dublin, Ireland. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Gavin Clark and Sarah Egan. 2015. The socratic method in cognitive behavioural therapy: A narrative review. *Cognitive Therapy and Research*, pages 1–17.

Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. Measuring conversational uptake: A case study on student-teacher interactions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1638–1653, Online. Association for Computational Linguistics.

Hoda Fakour and Moslem Imani. 2025. Socratic wisdom in the age of ai: a comparative study of chatgpt and human tutors in enhancing critical thinking skills. *Frontiers in Education*, 10.

Lucile Favero, Juan Antonio Pérez-Ortiz, Tanja Käser, and Nuria Oliver. 2024. Enhancing critical thinking in education by means of a socratic chatbot. *Preprint*, arXiv:2409.05511.

H Paul Grice. 1975. Logic and conversation, chapter syntax and semantics: Speech acts.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. Deepseek-coder: When the large language model meets programming – the rise of code intelligence. *Preprint*, arXiv:2401.14196.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-coder technical report. *Preprint*, arXiv:2409.12186.

Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. Learning to identify follow-up questions in conversational question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 959–968, Online. Association for Computational Linguistics.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *Preprint*, arXiv:2505.06120.

Nguyen-Thinh Le. 2019. How do technology-enhanced learning tools support critical thinking? *Frontiers in Education*, 4.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

M Maxwell and M Maxwell. 2014. How to use the socratic method. the socratic method research portal.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Phil Newton, Ana Da Silva, and Lee Peters. 2020. A pragmatic master list of action verbs for bloom's taxonomy. *Frontiers in Education*, 5:107.

Jae-Woo Park, Seong-Jin Park, Hyun-Sik Won, and Kang-Min Kim. 2024. Large language models are students at various levels: Zero-shot question difficulty estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8157–8177, Miami, Florida, USA. Association for Computational Linguistics.

Richard Paul and Linda Elder. 2008. Critical thinking: The art of socratic questioning, part iii. *Journal of Developmental Education*, 31.

Pagnarith Pit, Tanya Linden, and Antonette Mendoza. 2024. Generative artificial intelligence in higher education: One year later. In *AMCIS 2024 Proceedings*.

Chris Quintana, Brian J. Reiser, Elizabeth A. Davis, Joseph Krajcik, Eric Fretz, Ravit Golan Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. 2018. *A Scaffolding Design Framework for Software to Support Science Inquiry*, pages 337–386. Taylor and Francis, United States.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Marco T. Rodriguez Sandoval, Gianny M. Bernal Oviedo, and Maria I. Rodriguez-Torres. 2022. From preconceptions to concept: The basis of a didactic model designed to promote the development of critical thinking. *International Journal of Educational Research Open*, 3:100207.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, and 7 others. 2024. Code llama: Open foundation models for code. *Preprint*, arXiv:2308.12950.

Nikita Salkar, Thomas Trikalinos, Byron Wallace, and Ani Nenkova. 2022. Self-repetition in abstractive neural summarizers. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 341–350, Online only. Association for Computational Linguistics.

Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, ICER '22, page 27–43, New York, NY, USA. Association for Computing Machinery.

John R Searle. 1975. Indirect speech acts. In *Speech acts*, pages 59–82. Brill.

Kumar Shridhar, Jakub Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. Automatic generation of socratic subquestions for teaching math word problems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4136–4149, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Katherine Stasaski and Marti A. Hearst. 2017. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312, Copenhagen, Denmark. Association for Computational Linguistics.

Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada. Association for Computational Linguistics.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *Preprint*, arXiv:2205.07540.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024a. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024b. Large language models for education: A survey and outlook. *ArXiv*, abs/2403.18105.

Justine Zhang and Cristian Danescu-Niculescu-Mizil. 2020. Balancing objectives in counseling conversations: Advancing forwards or looking backwards. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5276–5289, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore:

Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Lily Zou, Alexander King, Salil Soman, Andrew Lischuk, Benjamin Schneider, David Walor, Mark Bramwit, and Judith K. Amorosa. 2011. Medical students' preferences in radiology education: A comparison between the socratic and didactic methods utilizing powerpoint features in radiology education. *Academic Radiology*, 18(2):253–256.

A Models' System Prompt

This section details the system prompt used to condition the behavior of both models to simulate a student during a programming dialogue.

Format Prompt This is used to maintain consistency in the responses of both models:

$format\_prompt$ = "Start your answer with Student:"

Context Prompt This is the behavioral prompt used to control the simulated behavior of a novice programming student.

$context\_prompt$ = " You are a first-year programming student who is learning how to debug your Python code. You are in the middle of a tutoring session. Your task is to respond to the latest question asked by your teacher in the dialogue so far. Your goal is to find a solution to the bug code through Socratic dialogue. Respond like a real student: - Think out loud and explain your reasoning. - Only respond to the most recent question. - Use the code you've been working on to guide your answer.

System Prompt We simply concatenate the two prompts together to create the final system prompt:

$SysPrompt = format\_prompt + context\_prompt$

B Mutual Information of Tokens in NextQ and Variant Classes

Table 4 represents the mutual information of in-class occurrence of tokens - ranked by top 10 in each class. The full list can be acquired by running the source code provided in the repository.

BEST			MEDIUM			WORST		
Feature	MI	Freq$_{pos}$	Feature	MI	Freq$_{pos}$	Feature	MI	Freq$_{pos}$
way	0.033620	0.089286	observe	0.019945	0.053571	correctly	0.047615	0.125000
loop	0.025526	0.125000	say	0.019945	0.053571	condition	0.038928	0.160714
let	0.025366	0.321429	print	0.019945	0.071429	adjust	0.033620	0.089286
python	0.021669	0.053571	value	0.016227	0.267857	modify	0.031842	0.125000
inputs	0.019945	0.053571	python	0.016216	0.142857	new	0.026743	0.071429
did	0.018689	0.035714	variables	0.015633	0.089286	returning	0.026743	0.071429
code	0.017397	0.250000	method	0.015574	0.035714	problem	0.025693	0.107143
example	0.015574	0.035714	return	0.013677	0.160714	loop	0.021419	0.214286
check	0.015466	0.017857	terminal	0.013223	0.035714	operator	0.019971	0.178571
line	0.014429	0.285714	examples	0.013223	0.035714	calculating	0.019945	0.053571

Table 4: Top 10 features for each class, ranked by mutual information (MI) along with their frequency