Fine-Tuning GPT-2 for Socratic Mathematics Tutoring on MathDial: Experimental Report

Chiș Bogdan-Mihai
Faculty of Mathematics and Informatics
Babeş-Bolyai University
Cluj-Napoca, Romania
bogdan.mihai.chis@stud.ubbcluj.ro

***Abstract*—This report documents the experimental component of a project that fine-tunes a GPT-2 model for Socratic-style mathematics tutoring using the MathDial tutoring dialogue dataset. The focus is on experimental design, dataset preparation, training configurations, evaluation protocols, and results analysis. The study investigates how dataset formatting, supervision targets, and decoding constraints influence (i) pedagogical behavior (questioning, hinting, and refusal to give full solutions) and (ii) task performance on dialogue-based tutoring turns. All artifacts (code, preprocessing scripts, and experiment configs) are intended to be released in a reproducible repository.**

I. Experimental Setup

A. Proposed Approach (Training Objective)

We fine-tune a GPT-2 language model to generate *tutor turns* conditioned on prior dialogue context. Each training sample is a tutoring episode segment consisting of alternating student and tutor utterances. The model is trained with standard causal language modeling (CLM): given a tokenized prompt containing the dialogue history, it predicts the next tutor response.

To encourage Socratic behavior rather than solution dumping, the training data is formatted to emphasize (a) tutor questioning moves, (b) incremental hints, and (c) short, targeted feedback. In addition, we evaluate a constrained decoding variant that discourages direct final-answer patterns (Section I-E).

B. Design of Experiments

The experiments are organized around three factors:

- **F1: Input formatting.** How the dialogue context is serialized (special tokens, role tags, turn separators) and the context window length.
- **F2: Supervision target.** Predicting *only* tutor turns vs. predicting the entire dialogue continuation and masking loss for student turns.
- **F3: Generation constraints.** Unconstrained sampling vs. constrained decoding that reduces probability of direct-solution language.

Table I summarizes the experimental grid. The goal is not to exhaustively search hyperparameters, but to compare meaningful configuration choices that directly impact tutoring behavior.

TABLE I
Experiment Grid (Configuration Factors)

ID	Training Target	Decoding
E1	Tutor-only (loss on tutor tokens)	Sampling (baseline)
E2	Tutor-only (loss on tutor tokens)	Constrained decoding
E3	Masked CLM (full context, masked loss)	Sampling
E4	Masked CLM (full context, masked loss)	Constrained decoding

C. Dataset: MathDial

MathDial is a collection of math tutoring dialogues in which a tutor guides a student through problem solving using pedagogically motivated moves (e.g., asking questions, giving hints, eliciting self-explanations) rather than directly providing final solutions [1]. Each dialogue contains alternating turns and, depending on the subset, may include metadata such as problem statements, tutor action labels, or move annotations.

1) Preprocessing and Splits: The dataset is cleaned and normalized as follows:

- standardize whitespace and remove corrupted examples;
- normalize role tags to a fixed schema (`<STUDENT>`, `<TUTOR>`);
- optionally truncate or window dialogue history to fit the context length;
- filter out samples where the tutor turn contains only a final answer (to reduce solution-leakage supervision).

We use the official train/validation/test splits when provided; otherwise, we apply an 80/10/10 split with stratification by dialogue length (short vs. long episodes) to keep distributions similar across splits.

D. Model and Training Configuration

1) Base Model: The base model is GPT-2 (small) initialized from a public checkpoint. The tokenizer is extended with special role tokens and separators, and embeddings are resized accordingly.

2) Optimization: Fine-tuning uses AdamW with a linear warmup followed by decay. Gradient clipping is applied to stabilize training. The following parameters are treated as the default configuration, and only changed when explicitly stated:

- batch size: 8–32 (depending on GPU memory), gradient accumulation to match an effective batch size;
- learning rate: $1e{-5}$ to $5e{-5}$;
- epochs: 3–5 with early stopping on validation loss;
- max sequence length: 512–1024 tokens, depending on the experiment.

3) Loss Masking Variant: For experiments E3–E4, we keep the full dialogue serialized but compute loss only on tutor segments. This preserves student context while preventing the model from learning to generate student answers as part of the objective.

E. Decoding Strategy

We compare:

- **Sampling baseline:** top-p nucleus sampling with a moderate temperature.
- **Constrained decoding:** a lightweight constraint that penalizes phrases associated with direct final-answer delivery (e.g., "the answer is", "therefore x =", "final answer") and prefers questions/hints. This is implemented as a decoding-time logit penalty over a small lexicon, keeping the model unchanged.

The constrained decoding is evaluated for pedagogical benefit and potential trade-offs in helpfulness or coherence.

II. Evaluation Methodology

A. Automatic Metrics

We evaluate tutor-turn generation with:

- **Perplexity (PPL)** on validation/test tutor tokens.
- **Token-level F1 / ROUGE-L (optional)** against reference tutor turns, reported cautiously because multiple tutor responses can be valid.

B. Pedagogical Behavior Metrics

Because tutoring quality is not captured by lexical overlap alone, we also measure:

- **Question rate:** fraction of generated tutor turns containing at least one question mark or interrogative pattern.
- **Hinting rate:** fraction of turns containing scaffolding markers (e.g., "try", "think about", "what if", "consider").
- **Solution leakage rate:** fraction of turns that explicitly provide the final numeric/symbolic result when the dataset annotation indicates the tutor should be guiding rather than concluding.

If MathDial provides teacher-move labels, we additionally compute accuracy/F1 for predicting move-consistent responses by mapping generated turns to move categories using a lightweight classifier (or pattern-based heuristic) as an auxiliary analysis.

C. Human Evaluation Protocol (Small-Scale)

A small human evaluation is conducted on a random sample of dialogues from the test set. Annotators rate each generated tutor turn on:

- **Socraticity:** encourages student thinking (questions, prompts, scaffolding).
- **Correctness:** mathematical validity of hints/feedback.
- **Helpfulness:** progress toward solving without giving away.
- **Coherence:** consistency with dialogue history.

Each criterion is rated on a 1–5 Likert scale.

III. Results

A. Quantitative Results

Table II is the main reporting table for automatic and behavior metrics. Values should be filled with measured results from the runs.

B. Ablation: Formatting and Context Length

To isolate the impact of context length, we additionally report a small ablation where the same model is trained with shorter vs. longer history windows. The expectation is that longer context improves coherence but may increase overfitting to solution patterns if not carefully filtered.

C. Qualitative Examples

We include representative dialogue snippets illustrating:

- effective scaffolding (good Socratic questions and hints),
- failure modes (hallucinated steps, premature final answers),
- how constrained decoding changes tutor behavior.

Examples are selected from the test set and anonymized if needed.

IV. Discussion

A. Interpretation of Results

We interpret improvements in pedagogical behavior metrics (question and hint rates, reduced leakage) alongside automatic metrics. Lower perplexity alone does not guarantee tutoring quality; therefore, the primary success criterion is a balanced outcome: coherent and mathematically correct tutor turns that guide the student without disclosing full solutions too early.

B. Observed Failure Modes

Common issues expected in GPT-2 tutoring fine-tuning include:

- **Solution dumping:** model outputs the final result immediately;
- **Shallow questioning:** repetitive or generic questions that do not advance reasoning;
- **Hallucinated math:** incorrect algebraic manipulation or arithmetic;
- **Context drift:** ignoring student errors or prior steps.

C. Reproducibility Considerations

We emphasize reproducibility by fixing random seeds, logging full configs, and versioning the dataset preprocessing scripts. We also report hardware and training time to contextualize resource requirements.

TABLE II

MAIN RESULTS ON THE MATHDIAL TEST SET (*placeholder numbers for paper formatting only; replace with measured results*)

Exp	**PPL** ↓	**ROUGE-L** ↑	**Q-Rate** ↑	**Hint-Rate** ↑	**Leakage** ↓	**Human (avg)** ↑
E1	22.8	0.247	0.54	0.46	0.18	3.4
E2	23.6	0.240	0.68	0.60	0.10	3.9
E3	20.9	0.265	0.58	0.52	0.15	3.6
E4	21.7	0.258	0.71	0.64	0.08	4.1

V. ARTIFACTS AND REPRODUCIBILITY

All artifacts are be made available in a public repository:

https://github.com/bogdan-chis/ai-math-tutor

The repository is intended to include:

- preprocessing scripts that convert MathDial to the training format;
- training and evaluation scripts with config files for E1–E4;
- a `README.md` with step-by-step instructions to reproduce results;
- exported checkpoints and/or inference instructions.

VI. THREATS TO VALIDITY

A. *Internal Validity*

Model outcomes may depend on preprocessing choices (filtering solution-like turns) and decoding heuristics. We mitigate this by explicitly documenting the pipeline and providing ablations.

B. *Construct Validity*

Automatic metrics such as ROUGE may not reflect pedagogical quality. We mitigate this with behavior metrics and small-scale human evaluation.

C. *External Validity*

Results on MathDial may not generalize to other domains or more advanced math curricula. Future work should validate on additional tutoring benchmarks and out-of-distribution problems.

REFERENCES

[1] J. Macina, N. Daheim, S. P. Chowdhury, T. Sinha, M. Kapur, I. Gurevych, and M. Sachan, “Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Dec. 2023, pp. 5602–5621. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.372/