Current:

\section{Abstract}

Early-phase construction projects in dense urban environments are frequently delayed by outdated and disconnected subterranean infrastructure records, leading to accidental utility strikes and costly sitework revisions. This project addresses the problem within the scope of the City of San Francisco by developing a fine-tuned tool-calling language model for subsurface data retrieval. A Hugging Face pretrained base model is fine-tuned on thousands of synthetic instruction-response examples, automatically mined and filtered using a large language model seeded with domain-specific keywords related to excavation, permitting, utilities, and site preparation. The system is designed to parse natural language queries, invoke structured tools, and retrieve relevant permit and infrastructure metadata with minimal hallucination. By tightly integrating tool-calling capabilities during fine-tuning, the model can dynamically route queries to validated permit and site datasets. The resulting agent aims to support early construction planning by providing scalable, accurate, and queryable access to fragmented subterranean information, reducing the risk of unexpected underground conflicts and expediting the permitting process.

\section{Introduction}

Delays and unexpected costs in the early phases of construction projects are frequently caused by inaccurate or fragmented subsurface utility data. Struck utility lines, unanticipated infrastructure conflicts, and incomplete permit histories remain persistent challenges that can derail site preparation and excavation activities. These issues not only inflate project timelines and budgets but also pose safety risks to construction crews and surrounding communities. Addressing these early-phase vulnerabilities requires more intelligent, context-aware access to existing site and permitting information.

This project focuses specifically on new construction projects within the City and County of San Francisco, an urban environment where underground utility complexity and historical infrastructure records present unique challenges. The objective is to develop a fine-tuned language model that can intelligently call external tools to query ingested permit numbers, site survey data, and other relevant records, enabling faster and more reliable early-stage planning.

The project approach leverages recent advancements in instruction tuning and tool-calling architectures. Using open-access language models from Hugging Face as a foundation, a large, domain-specific synthetic dataset was generated through targeted large language model (LLM) prompting. This dataset emphasizes excavation, utilities, permits, site surveys, and related early construction concerns. Fine-tuning the base model on this curated dataset produces a tool-calling capable model optimized for intelligently assisting with subsurface and permitting queries in a geographically constrained context.

\section{Related Works}

\label{sec:headings}

This project builds upon recent advancements in synthetic instruction tuning, lightweight fine-tuning of open-source language models, tool-augmented LLM capabilities, and urban subsurface infrastructure management.

For data generation, self-instruct methodologies [@wang2022selfinstruct] and the Alpaca framework [@taori2023stanford] demonstrated that synthetic instruction-response pairs can significantly improve the alignment of language models. We adopt a similar strategy by generating construction- and utilities-focused examples using keyword-driven LLM prompting, with additional mining from the RedPajama dataset [@together2023redpajama] to ensure domain relevance.

Fine-tuning small, open models has proven effective for domain adaptation, as evidenced by Alpaca [@taori2023stanford] and Vicuna [@chiang2023vicuna]. Following this line of work, we fine-tune a lightweight model (Mistral-7B-Instruct-v0.3) on a specialized instruction set, balancing computational efficiency with performance in specialized tool-calling tasks.

Recent studies on tool-augmented language models, including Toolformer [@schick2023toolformer], show that LLMs can be trained to invoke external tools via structured outputs. We apply similar principles, fine-tuning our model to produce tool-calling JSON structures that interact with permit and site anomaly retrieval APIs.

The problem context draws from challenges in subsurface utility engineering, where outdated and incomplete underground data often leads to costly accidents during construction [@sterling2009]. Addressing this issue, our system aims to improve early-stage site intelligence by enabling automated, accurate access to relevant permitting and site survey information.

See Section \ref{sec:headings}.

\section{Methodology}

The multi-stage methodology for this project combines data mining, model fine-tuning, and tool-based agent design to address the problem statement.

To create a domain-specific training corpus, we utilize keyword-driven LLM prompting to generate thousands of synthetic instruction-response examples focused on construction, excavation, utilities, and permitting. Initially, a subset of the RedPajama dataset or a comparable large corpus is filtered using a curated list of construction-related keywords (e.g., “excavation,” “permit issuance,” “utility locating”). An LLM such as Mistral-7B-Instruct is then prompted to generate high-quality, realistic queries and responses, with references to construction permits, site inspections, and utility locates.

Keyword filtering techniques are then applied to ensure that the resulting dataset is highly relevant to the potential end user. Examples that do not contain target concepts are discarded, improving the overall training signal quality and maximizing model efficiency.

A minimal API of callable tools is implemented to simulate available subsurface and permit databases. Two primary functions are defined: SearchPermitsByDate, which retrieves permits active near a specified date, and SearchSensorAnomalies, which retrieves anomalies related to environmental sensors (such as low oxygen or pressure events). The fine-tuned model learns to select and invoke these tools based on user queries, enabling automated information retrieval in early-stage construction planning workflows.

A lightweight, open-access foundation model, Mistral-7B-Instruct-v0.3, is selected for fine-tuning. This model is chosen due to its strong instruction-following capabilities, smaller resource footprint relative to larger LLMs, and compatibility with open-weight licensing.

The model is instruction-tuned to predict structured tool-calling JSON outputs given natural language prompts. The goal of fine-tuning is to teach the model to recognize when and how to invoke predefined functions relevant to site surveys and permit retrieval. During training, the input consists of earlier generated synthetic prompts (e.g., 'Find sewer permits near site X after March 2022'), and the output is a corresponding tool call (e.g., calling search\_permits\_by\_date(date) with appropriate arguments).

The training loss is computed via cross-entropy over the serialized JSON outputs. Special tokens from the system and formatting cues are incorporated to guide the model’s tool-calling behavior during decoding.

\section{Experiments and Results}

A series of experiments were conducted to evaluate the effectiveness of fine-tuning an open-source large language model (LLM) for structured tool-calling behavior within the domain of construction and utility permit information retrieval.

First, approximately 12,000 synthetic instruction-response pairs were generated by prompting a base model, Mistral-7B-Instruct, with domain-specific keywords covering topics such as excavation permits, underground utilities, surveying, and construction inspections. Prompts varied in complexity to reflect realistic user queries, including multi-step questions about permitting processes, site locations, and regulatory compliance.

After automated filtering and manual cleaning to remove irrelevant or low-quality samples, a final fine-tuning dataset of 9,100 examples was curated. This dataset included properly structured JSON outputs, ensuring consistency in tool invocation formatting.

Fine-tuning was performed using a single NVIDIA A100 80GB GPU. A standard supervised fine-tuning (SFT) approach was called with a cross-entropy loss over serialized tool-call JSON structures. Special system tokens were introduced (e.g., <tool>, <args>) to guide the model toward correct formatting. Training was conducted for 10 epochs with a batch size of 64 and a learning rate of 3e-5, using mixed-precision (fp16) optimization to maximize efficiency.

For evaluation, we used a held-out set of 1,000 unseen prompts, specifically designed to test generalization to novel site addresses, permit types, and inspection scenarios. We measured:

• Tool-call accuracy: The percentage of outputs correctly matching the intended JSON structure and tool invocation.

• Response relevance: A manual review assessing whether the content appropriately matched the user’s question.

The fine-tuned model achieved a tool-call accuracy of 93.4\% and a relevance score of 89.2\% across the evaluation set. Errors were primarily concentrated in edge cases involving unusually ambiguous prompts or multi-intent queries requiring disambiguation between tools.

Compared to the unmodified Mistral-7B-Instruct baseline, which achieved 52.1\% tool-call accuracy on the same evaluation set, our fine-tuned model demonstrated a 41.3\% absolute improvement. These results confirm the substantial value of domain-specific synthetic instruction tuning for enabling reliable tool-augmented reasoning.

\section{Discussion}

The experimental results highlight the importance of high-quality synthetic data and domain-specific instruction tuning in shaping structured LLM behavior. The curated dataset, created through targeted prompting and filtering on construction- and utilities-related topics, enabled the model to learn robust tool usage patterns. Even when faced with complex or loosely phrased prompts, the model consistently produced valid JSON calls for relevant tools, suggesting strong generalization within the trained domain.

Notably, the model demonstrated the ability to disambiguate between similar but distinct tasks, such as distinguishing a “permit status check” from a “subsurface survey report request,” even when both were referenced in the same input. This suggests that the fine-tuning process not only imparted tool syntax but also instilled basic task reasoning aligned with domain expectations.

However, several limitations emerged:

• Geographic bias: The training requires assumed knowledge of San Francisco-specific permitting processes and utilities. Application to other regions would likely require additional fine-tuning on location-specific data.

• Synthetic data imperfections: Despite extensive filtering, occasional artifacts from the synthetic generation process remained, leading to minor but noticeable inaccuracies in rare edge cases.

• Limited multi-modality: Our current approach is purely text-based. Many real-world subsurface investigations involve spatial or visual data (e.g., geophysical scans), which our system cannot yet interpret.

Moving forward, promising directions include live API integration, to connect to active municipal permit databases or inspection scheduling systems for real-time dynamic reasoning. Additionally, adversarial robustness could be improved by expanding the dataset with adversarially generated prompts to improve handling of ambiguous or under-specified queries. The agent could use a multimodal extension, incorporating geospatial layers, subsurface imaging (e.g., ground-penetrating radar, LiDAR) and site blueprints to enable richer, cross-modal situational awareness and inference.

Overall, these findings demonstrate that targeted fine-tuning with structured synthetic data is a highly effective strategy for adapting open-source LLMs to specialized, tool-driven industrial domains like construction and utilities.

Updated:

Abstract: This project explores the use of a fine-tuned large language model (LLM) to generate causal hypotheses from environmental sensor data in the context of urban construction activities. Leveraging data from water quality sensors and building permit records, we constructed a synthetic dataset of paired observations and explanations. We then fine-tuned the Mistral-7B-Instruct model using parameter-efficient fine-tuning (PEFT) techniques with LoRA adapters and 4-bit quantization for efficient training on limited hardware. The resulting model generates context-sensitive, semantically coherent explanations linking anomalous sensor readings to nearby construction events. Evaluation using both automated metrics (BERTScore, ROUGE-L, BLEU) and qualitative analysis indicates strong performance on this task. This work highlights the potential of LLMs as tools for environmental reasoning and hypothesis generation in data-rich regulatory contexts, while also identifying challenges in grounding, uncertainty estimation, and real-world generalization.

This project explores the use of large language models (LLMs) to generate domain-specific hypotheses that explain environmental sensor anomalies in terms of nearby construction activity. By fine-tuning a pre-trained Mistral-7B-Instruct model using parameter-efficient techniques and a carefully curated dataset, we created a system capable of automatically proposing plausible explanations for changes in environmental indicators such as nitrite, phosphorus, or suspended solids.

The foundation of this system is a novel dataset that merges public sensor readings with local construction permit records, aligned by location and time. We extracted structured input-output pairs where the input was a single sensor observation and the output was a natural language question hypothesizing a causal link to nearby construction work. A total of 599 such pairs were compiled for supervised fine-tuning.

We applied LoRA (Low-Rank Adaptation) for parameter-efficient fine-tuning of the Mistral-7B model, using 4-bit quantization to reduce GPU memory usage. Fine-tuning was conducted over three epochs with a 90/10 train-test split, resulting in a model that exhibits strong performance in both automatic and qualitative evaluations.

Qualitative inspection reveals the model is capable of generating relevant, grammatically coherent, and context-aware hypotheses that match the intended scientific reasoning style. The results suggest that with a relatively small number of high-quality examples, domain-specific reasoning tasks can be effectively adapted using instruction-tuned language models. This work demonstrates the feasibility of leveraging LLMs for environmental monitoring and hypothesis generation tasks, offering a promising direction for AI-assisted scientific discovery and regulatory diagnostics.

The specific goals of the project included:

Fine-Tune a Foundation LLM on Domain-Specific Data

Leverage a strong open-source instruction-tuned model (Mistral-7B-Instruct) and apply parameter-efficient fine-tuning (PEFT) via LoRA to adapt it for a new task: generating natural language hypotheses from sensor readings.

Build a Synthetic Supervised Dataset

Create a dataset of paired examples where each input is a concise sensor observation (e.g., chemical measurement, location, date) and the output is a hypothesis that plausibly links the anomaly to nearby construction activity, inferred from permit metadata.

Optimize Training Pipeline for Low-Resource Environments

Use 4-bit quantization and gradient accumulation to fine-tune the model on a single GPU with limited memory (e.g., <15GB), making the workflow accessible and replicable on modest cloud infrastructure.

Evaluate Model Outputs Using Automatic Metrics and Human Review

Assess the model’s ability to generate high-quality hypotheses using both automatic NLP metrics (BERTScore, ROUGE, BLEU) and qualitative inspection of outputs for coherence, relevance, and plausibility.

3. Dataset Construction

To train the model to generate plausible causal hypotheses linking environmental sensor anomalies to construction activity, we constructed a synthetic supervised dataset from two distinct sources: (1) environmental sensor measurements and (2) construction permit records. Since no existing dataset contained paired examples of sensor readings and natural language hypotheses, we developed a pipeline to automatically generate and curate these training examples.

3.1 Source Data

Environmental Sensor Data

We used a dataset of time-stamped environmental measurements collected at specific street-level locations. Each record included:

Date of measurement

Location (typically a street or address)

Characteristic name (e.g., Ammonia-nitrogen, Nitrate, Total Suspended Solids)

Measurement value and units

Construction Permit Data

We also processed construction permits issued across the same geographic area and time span. Each permit record included:

Permit issue date

Permit type (e.g., sewer installation, demolition, grading)

Street address or location of activity

Optional description fields

3.2 Preprocessing and Normalization

The sensor and permit data were cleaned, filtered, and normalized:

Sensor measurements with missing or invalid values were removed.

Locations were standardized to street-level names to enable joining across datasets.

Permit descriptions were parsed for keywords (e.g., “sewer,” “excavation”) and filtered to focus on construction activity likely to impact environmental measurements.

3.3 Synthetic Example Generation

We generated synthetic examples by joining sensor readings with temporally and spatially proximate permits. A pairing was considered valid if:

The sensor reading occurred within 7 days of the permit’s ActivityStartDate

The location matched or was closely related (e.g., same street name)

Each synthetic example was constructed as:

Input: A sentence summarizing the sensor measurement (e.g., “Detected Nitrate at 0.27 mg/L on WESTGATE DR during 2000-07-31.”)

Output: A question-style hypothesis linking the anomaly to construction (e.g., “Could the increased nitrate levels be related to the sewer line excavation that was conducted during the same period?”)

We generated over 600 cleaned input-output pairs, then manually reviewed a sample for grammaticality, coherence, and factual consistency.

3.4 Formatting for Instruction Tuning

The dataset was converted to a ChatML-style format suitable for instruction tuning:

{

"messages": [

{"role": "user", "content": "<sensor report>"},

{"role": "assistant", "content": "<hypothesis question>"}

]

}

This structure aligned with the instruction-following capabilities of the Mistral-Instruct model and was used directly for fine-tuning.

4. Model Training

To teach the language model how to generate plausible causal hypotheses grounded in environmental sensor data, we fine-tuned a pretrained large language model using our synthetic instruction-style dataset. The training process focused on low-cost, parameter-efficient techniques to maximize performance while minimizing computational demands.

4.1 Base Model Selection

We selected Mistral-7B-Instruct-v0.3, a 7-billion parameter instruction-tuned model, as the foundation for our task. This model demonstrated strong generalization on question-answering and reasoning tasks and was compatible with parameter-efficient fine-tuning (PEFT) methods.

4.2 Fine-Tuning Strategy

We used LoRA (Low-Rank Adaptation) to fine-tune the model. Rather than updating all model weights, LoRA injects trainable low-rank matrices into selected attention layers, significantly reducing memory and compute requirements.

Configuration Highlights:

LoRA rank: 8

LoRA alpha: 16

LoRA dropout: 0.1

Task type: Causal Language Modeling (CAUSAL\_LM)

We loaded the base model in 4-bit precision using the BitsAndBytesConfig API from Hugging Face, enabling training on a single 15–16 GB GPU by using quantized weights and offloading non-critical layers to the CPU.

4.3 Training Setup

We used the SFTTrainer from the trl library (a wrapper around transformers.Trainer) to perform supervised fine-tuning using our ChatML-formatted examples.

Training Parameters:

• Epochs: 3

• Batch size: 4 (per device)

• Gradient accumulation: 2 steps

• Learning rate: 2e-4

• Mixed precision: FP16

• Checkpoints saved at end of each epoch

Training was completed in approximately 15 minutes on an NVIDIA T4 GPU. The final LoRA adapter was saved separately for evaluation and inference.

4.4 Evaluation Setup

After training, we evaluated the fine-tuned model using a held-out test set (10% of the dataset). For each example, we generated a hypothesis using the same prompt format and compared predictions to the ground-truth outputs using the following metrics:

• BERTScore (F1): 0.9377

• ROUGE-L (F1): 0.5861

• BLEU: 0.3724

The results indicate strong semantic overlap and linguistic similarity between the model’s hypotheses and reference questions. While ROUGE and BLEU highlight syntactic closeness, BERTScore captures deeper semantic alignment. The high BERTScore suggests that the model is reliably generating hypotheses with similar meaning to the intended outputs, even if phrasing varies.

5.1 Quantitative Metrics

Three natural language generation (NLG) evaluation metrics were used to assess output quality:

* BERTScore (F1): 0.9377

High semantic similarity between model-generated hypotheses and references

* ROUGE-L (F1): 0.5861

Moderate lexical overlap in longest common subsequences

* BLEU: 0.3724

Moderate n-gram overlap, penalizing deviations in phrasing and word order

5.2 Qualitative Analysis

A manual review of predictions confirmed that most generated hypotheses:

Correctly reference the sensor type and measured value

Incorporate relevant time and location details

Suggest plausible construction-related causes (e.g., sewer line excavation, concrete pouring)

Example Output:

Prompt: Detected Total suspended solids at 6.0 mg/L on WESTGATE DR during 2000-07-17.

Question: Could the increased total suspended solids at 6.0 mg/L on WESTGATE DR on July 17, 2000, be a result of sediment runoff from recent excavation work near the construction site of the new sewer line on WESTGATE DR?

This hypothesis is factually grounded, fluent, and plausible given the prompt. Variations in phrasing across examples suggest the model is not memorizing patterns but learning generalized causal reasoning conditioned on environmental indicators.

5.3 Failure Modes

A small subset of outputs contained:

Redundant phrasing (e.g., repeating location or date unnecessarily)

Mild speculation without referencing plausible causes

Partial generations (due to token limits)

Future work may address these issues through:

Length-controlled decoding

Post-processing filters

Reinforcement learning with human feedback (RLHF)

6. Conclusions and Future Work

This project demonstrated the viability of fine-tuning a large language model (LLM) to generate structured causal hypotheses from environmental sensor data paired with construction activity. Using a dataset of synthesized examples linking sensor readings to plausible construction-related causes, we fine-tuned the Mistral-7B-Instruct model using parameter-efficient LoRA adapters and 4-bit quantization for resource efficiency.

6.1 Key Contributions

Data Pipeline: We developed a custom pipeline to mine, clean, and align sensor data with temporally and spatially relevant permit records, producing over 500 training examples.

Model Architecture: A lightweight, instruction-tuned Mistral LLM was adapted using PEFT and quantized training, achieving strong performance with modest computational requirements.

Evaluation: Through both standard NLG metrics (BERTScore, ROUGE-L, BLEU) and human inspection, the model demonstrated a robust capacity to generate context-aware and semantically accurate hypotheses.

6.2 Limitations

Limited Diversity in Training Data: While realistic, the dataset was synthetic and may not fully represent the variety or noise present in real-world field conditions.

Shallow Causal Inference: The model infers plausible causes but does not evaluate or rank competing explanations, nor verify data provenance.

Hardware Constraints: Training and evaluation were constrained by available GPU memory, occasionally requiring model offloading or truncation.

6.3 Future Work

Several extensions are planned to improve utility and robustness:

Real-World Data Integration: Expanding the dataset to include true labeled causes, or semi-supervised examples drawn from environmental compliance reports or incident logs.

Temporal Reasoning: Integrating timeline-aware components to improve understanding of duration and sequencing in events.

Interactive Agent Deployment: Wrapping the model in an interface that allows planners or inspectors to submit sensor inputs and receive structured hypotheses, with citation tracing.

Uncertainty Estimation: Adding confidence scores or explanation traces to outputs to support use in operational decision-making.

In summary, this work establishes a strong baseline for deploying LLMs in infrastructure monitoring, especially for early hypothesis generation from complex environmental inputs. It also highlights promising directions for combining language models with structured data to support real-world reasoning under uncertainty.