Building a Model to Classify Electric Utility Field Hazard Observations

Vyom Dave
dave.107@osu.edu

Ayham Huq
huq.19@osu.edu

Abby Atchley
atchley.23@osu.edu

Abstract

This project develops a transformer-based classifier, informed by the Edison Electric Institute's Safety Classification and Learning (eeiSCL) Model, to accurately detect and categorize high-risk safety observations from electric utility field reports. Leveraging NLP and domain-specific hazard definitions, our approach aims to improve safety prioritization and prevention efforts for utility field worker's operations.

1 Introduction

Electric utility field workers often operate in hazardous conditions while maintaining and restoring power lines, especially during storm events and other emergencies. Ensuring their safety hinges on accurately identifying high-risk observations from large volumes of field reports. While our previous attempt leveraged Facebook's FAISS library for quick similarity search over these observations, we now seek to develop a more sophisticated classifier to detect and categorize specific types of hazards, drawing on the Edison Electric Institute's Safety Classification and Learning (eeiSCL) Model. By integrating Natural Language Processing (NLP) techniques with domain-specific knowledge from the eeiSCL Model, we aim to better capture the nuances of "high-energy" and "potentially serious" incidents, thereby improving how safety managers prioritize follow-up and preventive measures.

2 Data and Resources

Our primary dataset consists of approximately 20,000 real-world safety observations (freeform text comments) collected during "CORE visits." These comments vary in length and detail, typically describing near-misses, safety checks, and onsite conditions. Each record includes:

- **Date**: Timestamp of the observation.
- **Observation Type**: Categorical Label indicating a general safety theme ("fall protection," "electrical hazard").
- **Comments**: Freeform text from supervisors describing what they observed.

Additionally, we will use the **eeiSCL Model** (Hallowell et al., 2023) as a conceptual foundation, which categorizes hazards based on the presence of high-energy sources (e.g., voltage above 50V, falls from height over four feet, heavy rotating machinery, and so on) and examines whether direct controls (e.g., lockout/tagout) were in place. This model helps define key dimensions of seven different categories of incidents listed below:

- High Energy Serious Injury or Fatality (HSIF)
- Low Energy Serious Injury or Fatality (LSIF)
- Potential Serious Injury or Fatality (PSIF)
- Capacity
- Exposure
- Success
- Low Severity

These categories can be further subdivided into 3 tiers indicating the importance or learning potential of the incident. These tiers can be seen in Figure 1.

3 Previous Approach

In our previous approach, we used the JXM/CDE-small-v1 sentence transformer model to generate text embeddings for each safety observation. These embeddings were indexed with the Facebook AI's FAISS library, applying L2 normalization to compute similarity scores. We then created and maintained manual keyword lists for different categories of hazards, matching new observations with these

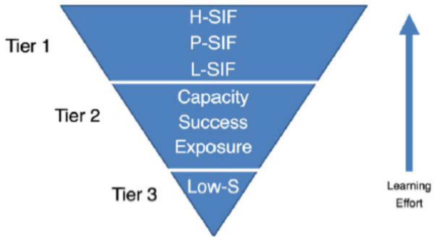

Figure 1: Hierarchical representation of incident categories organized in tiers (Note there is no hierarchy within the tiers)

lists through the FAISS similarity pipeline. Although this method produced satisfactory results and allowed us to group observations by thematic similarity, its reliance on fixed keywords and relatively shallow semantic matching left significant room for improvement.

4 Current Approach Overview

4.1 Data Preprocessing

To prepare our data set for supervised modeling, we first address the labeling of 20,000 unlabeled safety observations. Each observation must be categorized into one of seven eeiSCL classes—HSIF, PSIF, LSIF, Capacity, Success, Exposure, or Low Severity. We will leverage OpenAI's API service and curate a prompt to automatically generate labels for a portion of the 20,000 samples. This prompt will provide relevant context from the eeiSCL Model, ensuring that the system understands each category's definitions.

Next, we will subsample 50 observations from the 20,000 and manually annotate them according to domain expertise in electric-utility hazards. Comparing these ground-truth labels against the model-generated labels will give us an error metric, highlighting misclassifications and guiding prompt refinements. Our initial accuracy goal is 80% or higher; If the model underperforms, we will iterate by adjusting the prompt details or switching to a different OpenAI model, repeating the manual verification process until we exceed the accuracy threshold. Throughout, we will also perform basic text cleaning (e.g., removing extraneous whitespace) to ensure input consistency while preserving the key phrases that indicate the presence or absence of high-energy hazards and safety controls.

4.2 Model Fine-Tuning

For the fine-tuning stage, we will use two pretrained transformers from the DeBERTa family—**deberta-v3-base** (184M parameters) and **deberta-v3-large** (434M parameters). This selection enables a direct comparison between models of different parameter sizes within the same architecture family, allowing us to evaluate the tradeoffs between model complexity and performance for this specific domain. DeBERTa models are particularly well-suited for this classification task due to their enhanced disentangled attention mechanism and relative position encoding, which excels at capturing nuanced relationships in technical terminology common in electric utility observations. Additionally, these models have demonstrated state-of-the-art performance across multiple NLP benchmarks and are expected to effectively understand domain-specific language and complex safety descriptions through their advanced contextualized word embeddings.

Both models will be initialized with their default pretrained weights to leverage the linguistic knowledge gained from extensive pretraining on large text corpora, which improves sample efficiency and overall performance compared to random initialization. We will then adapt each model to our seven-category classification task using the labeled dataset generated in the preprocessing phase. During fine-tuning, we will use standard multi-class classification approaches (e.g., cross-entropy loss) and track various performance metrics. By comparing results across these two model variants, we aim to identify the balance between efficiency and predictive accuracy for capturing high-energy hazards and safety outcomes in real-world electric utility observations.

4.3 Error Analysis

To ensure a thorough evaluation of model performance, we will conduct a detailed error analysis on the classification results from both deberta-v3-base and deberta-v3-large. The goal is to identify common misclassification patterns and assess the models' strengths and weaknesses as part of the final evaluation.

4.3.1 Misclassification Scheme

Misclassifications will be categorized based on the following criteria:

- **Class-Level Misclassification:** Given the seven-category classification scheme, we will analyze cases where the model confuses one category for another, identifying frequently misclassified pairs.

- **Tier-Level Misclassification:** Since the seven categories are further organized into three tiers of varying importance, we will assess whether the model correctly assigns instances to the appropriate severity tier, even if the specific category is incorrect.

4.3.2 Qualitative Analysis

We will manually inspect a sample of misclassified instances to determine whether the misclassification is due to ambiguous wording in the input text and if the model struggles with certain linguistic patterns (e.g., negation, domain-specific terminology). We will also identify specific category pairs that the model confuses most often, assessing possible semantic overlap, and whether misclassifications tend to occur more frequently within or across severity tiers.

4.3.3 Quantitative Analysis

We will compute statistics to perform a quantitative assessment of error patterns, including precision, recall, and F1-score for each of the seven categories to measure model performance. We will also construct confusion matrices to visualize class-level misclassification trends and tier-level misclassification frequencies.

5 Methodology & Results

5.1 Data Preprocessing

We developed an automated labeling approach using the OpenAI GPT-4o-mini model to efficiently label 5,000/20,000 safety observations. Our implementation, detailed in the `OpenAISafetyClassifier` class, features a system for batch processing with built-in rate limiting and error handling to ensure reliable operation at scale.

5.1.1 Classification Methodology

Our implementation follows a systematic four-question decision framework derived from the eeiSCL Model:

1. Was high energy present?

2. Did a high energy incident occur?

3. Was a direct control present?

4. Was a serious injury sustained?

HE Present?	**HE Incident?**	**DC Present?**	**SI Sustained?**	**Class**
Yes	Yes	No	Yes	HSIF
Yes	Yes	No	No	PSIF
No	No	No	Yes	LSIF
Yes	Yes	Yes	No	Capacity
Yes	No	Yes	No	Success
Yes	No	No	No	Exposure
No	No	No	No	Low-Sev.

Table 1: Classification matrix based on the four-question framework (HE=High Energy, DC=Direct Control, SI=Serious Injury)

Each observation is processed through a carefully crafted prompt that includes precise definitions of key terms:

- **High Energy**: An element of work involving more than 500 ft-lbs of physical energy (e.g., lifting loads exceeding 500 pounds, working at heights above 4 feet, exposure to substances over 150°F)

- **Direct Control**: A barrier specifically targeted to mitigate exposure to high-energy sources

- **High-Energy Incident**: An instance where high-energy was released and a worker came in contact with or proximity to it

- **Serious Injury or Fatality**: Life-threatening or life-altering incident as defined by the EEI SIF Criteria

5.1.2 Classification Matrix

The classification of observations follows a deterministic mapping based on the combination of Yes/No answers to the four questions, as shown in Table 1.

5.1.3 Technical Implementation

Initially, we implemented a direct classification method using GPT-4o-mini with 4-shot learning examples, where the model was asked to classify observations directly into one of the seven categories. This approach achieved only 42% accuracy when compared to ground truth labels.

We then redesigned our approach to use the four-question framework and classification matrix described earlier. This structured approach significantly improved accuracy to 91%, as shown in Table 2.

Classification Approach	Accuracy
Direct classification with 4-shot examples	42%
Four-question framework with classification matrix	91%

Table 2: Accuracy of different labeling approaches compared to human expert judgments

Our implementation uses concurrent processing to label observations while adhering to API rate limits:

- ThreadPoolExecutor with 20 concurrent workers for parallel processing
- Exponential backoff retry logic for API errors
- Token tracking with 90% safety margin (200,000 tokens/minute limit)
- Batch processing (100 observations) with checkpoint saves

This approach enabled processing of 5,000 observations while maintaining high classification quality.

5.2 Model Fine-Tuning

After successfully generating high-quality labels through our four-question framework, we proceeded to fine-tune multiple transformer-based models on the labeled dataset. We split our 5,000 labeled observations into a training set of 2,500 observations and a development set of 2,500 observations. Additionally, we maintained a separate test set of 100 manually labeled observations to obtain evaluation against human judgments.

5.2.1 Training Process

We implemented two distinct fine-tuning approaches:

BERT Classification Model. Following best practices for fine-tuning transformer models (Heyamit, 2022), we implemented DeBERTa model training with a layer-freezing strategy. We froze all layers except the classification head, which significantly reduced training time while maintaining performance. The models were fine-tuned for both single-label classification (directly predicting one of seven categories) and multi-label classification (predicting answers to the four questions, then deriving the final category). We optimized using AdamW with a learning rate of 3e-5 and batch size of 8, with implementation details adapted from established fine-tuning guidelines. To address class imbalance, we implemented two specialized techniques:

- **Class weighting:** We applied partial class weights in the loss function, scaling the importance of minority classes.
- **Balanced sampling:** We utilized partial upsampling of minority classes, bridging approximately 50% of the gap between minority and majority classes.

LoRA Fine-tuning. Following implementation guidelines from the Lightning AI community resources (Lightning AI, 2023), we applied Low-Rank Adaptation (LoRA) on Mistral-7B-Instruct-v0.1. This parameter-efficient fine-tuning approach modified only a small subset of model parameters, specifically targeting query and value projection matrices with rank r=8 and alpha=16. The model was quantized to 4-bit precision to enable efficient training while preserving performance. We formatted inputs as structured prompts containing the incident text followed by our four classification questions.

Model Configuration	EM Accuracy
DeBERTa-v3-base (direct classification)	43%
DeBERTa-v3-base (four-question approach)	52%
DeBERTa-v3-large (direct classification)	63%
DeBERTa-v3-large (four-question approach)	63%
Mistral-7B-Instruct + LoRA (four-question)	69%

Table 3: Exact Match accuracy across model configurations on the manually labeled test set.

5.2.2 Experimental Results

Our experiments tested five model configurations, as shown in Table 3.

5.2.3 Discussion

Our experimental results reveal several key insights:

The four-question approach consistently outperformed direct classification with the base model, improving accuracy by 9 percentage points (43% to 52%). This indicates that breaking down the complex classification task into simpler binary decisions better aligns with the underlying logic of the eeiSCL Model.

However, the performance gap between the two approaches disappeared with the larger DeBERTa model, suggesting that increased model capacity can compensate for task complexity in direct classification. Nevertheless, we observed signs of overfitting with longer training epochs on the larger model, particularly for minority classes.

The most effective approach was the LoRA-tuned Mistral model using the four-question framework, which achieved 69% accuracy. This superior performance can be attributed to three factors: (1) the instruction-following capabilities of Mistral, (2) LoRA's effective parameter adaptation in the context of class imbalance, and (3) the structured nature of the four-question prompt, which guides the model through the classification decision tree.

One of the most critical issues we addressed was class imbalance, as the majority of observations fell into the "Low Severity" category. Our partial upsampling technique proved essential for improving performance, particularly for the DeBERTa models. Without balance adjustment, these models tended to default to predicting the majority class, significantly reducing their utility for identifying high-risk scenarios.

Interestingly, simply using a larger model did not guarantee better performance, showing that training methodology is more important than raw parameter count for specialized classification tasks our domain.

5.3 Error Analysis

The results of the best performing model, Mistral 7B, were further analyzed.

5.3.1 Qualitative Analysis

Examination of the Mistral 7B model's predictions (70.33% accuracy) reveals a systematic bias toward Low-Severity classifications. All misclassifications occurred at the tier level, with the model incorrectly labeling PSIF, Success, and Exposure instances as Low-Severity—categories distinguished by the presence of high energy. No significant linguistic patterns or predictive keywords were identified across safety classes; observations typically contained similar descriptive elements regardless of classification.

The determination of high energy presence depends on subtle semantic content rather than identifiable textual markers. Despite implementing both partial upsampling and class weighting techniques, the model's persistent tendency toward Low-Severity predictions suggests the challenge lies in recognizing implicit high-energy indicators without explicit hazard terminology in the text.

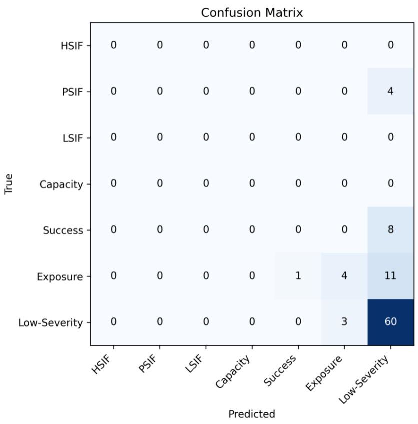

Figure 2: Confusion matrix for the LoRA-tuned Mistral model showing predictions across the seven safety categories.

5.3.2 Quantitative Analysis

The LoRA model achieved an overall accuracy of 70.33% and a weighted F1 score of 63.02% on our manually labeled test set. While these metrics indicate solid performance, our confusion matrix analysis (Figure 2) reveals several important patterns.

Most notably, the model exhibits a strong bias toward predicting the majority class (Low-Severity), with 83 out of 91 total predictions (91.2%) falling into this category. This conservative prediction pattern resulted in:

- All PSIF instances (4) being misclassified as Low-Severity
- All Success instances (8) being misclassified as Low-Severity
- Only 4 out of 16 Exposure instances (25%) being correctly identified

This heavy skew toward Low-Severity predictions, while contributing to higher overall accuracy due to class imbalance, is problematic for practical safety applications. The model correctly identifies Low-Severity cases with high precision (72.3%) but struggles significantly with recall for higher-risk categories.

From a safety perspective, this pattern indicates that the model is more prone to Type II errors (missing actual high-risk scenarios) than Type I errors (false alarms). While avoiding false alarms might be preferable in some contexts, the safety-critical nature of electric utility field operations suggests that a more balanced error profile would be beneficial, even at the cost of some precision.

The model's reluctance to classify observations outside of Low-Severity likely stems from two factors: (1) the persistent class imbalance in the training data despite our balancing efforts, and (2) the inherent challenge of identifying subtle indicators of high-energy situations in text descriptions that may lack explicit hazard terminology.

Future work should explore more aggressive balancing techniques, specifically targeting improved recall for the PSIF and Success categories, which represent significant learning opportunities for safety improvement despite their low base rate in the dataset.

5.4 Contributions

Ayham implemented the dataset labeling pipeline and LoRA fine-tuning approach. Vyom developed the BERT model fine-tuning methodology. Abby conducted the error analysis.

Each author wrote the sections corresponding to their technical contributions. Parts of this report were written with AI assistance, but all experimental design, implementation, analysis, and conclusions represent the authors' original work.

6 References

- Hallowell, M. R., Alexander, D., & Gambatese, J. A. (2017). Energy-based safety risk assessment: does magnitude and intensity of energy predict injury severity? Construction Management and Economics, 35(1–2), 64–77. https://doi.org/10.1080/01446193.2016.1274418

- Edison Electric Institute (EEI). (2023). *Safety Classification and Learning (SCL) Model.* https://www.eei.org/-/media/Project/EEI/Documents/Issues-and-Policy/Power-to-Prevent-SIF/eeiSCLmodel.pdf

- Chami, I., Abu-El-Haija, S., Perozzi, B., & Re, C. (2020). *Machine Learning on Graphs: A Model and Comprehensive Taxonomy*. https://doi.org/10.48550/arXiv.2005.03675

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* (p./pp. 5998–6008).

- Facebook AI Research. (2021). *FAISS library* https://github.com/facebookresearch/faiss

- Heyamit. (2022). *Fine-Tuning BERT for Classification: A Practical Guide*. Medium. https://medium.com/@heyamit10/fine-tuning-bert-for-classification-a-practical-g

- Lightning AI. (2023). *LoRA Insights: A Comprehensive Guide to Low-Rank Adaptation*. Lightning AI Community. https://lightning.ai/pages/community/lora-insights/