Journal of INTELLIGENT CONSTRUCTION

Research Article
https://doi.org/ 10.26599/JIC.2024.9180039

Automatic Construction Accident Report Analysis Using Large Language Models (LLMs)

Ehsan Ahmadi, Shashank Muley, Chao Wang(✉)

Bert S. Turner Department of Construction Management, Louisiana State University, Baton Rouge 70803, USA

Received: 2024-04-10 ***Revised:*** 2024-05-26 ***Accepted:*** 2024-06-05

ARTICLE INFO

Keywords

Construction Safety
Accident Report Analysis
Large Language
Models
GPT
Gemini
LLaMA

ABSTRACT

Construction site safety is a paramount concern, given the high rate of accidents and fatalities in the sector. This study introduces a novel approach to analyzing construction accident reports by employing advanced Large Language Models (LLMs), specifically GPT-3.5, GPT-4, Gemini Pro, and LLaMA 3. Our research focuses on the classification of key attributes in accident reports: root cause, injury cause, affected body part, severity, and accident time. The results reveal that GPT-4 achieves significantly higher accuracy across most attributes. Gemini Pro demonstrates superior performance in the "Injury Cause" classification, while LLaMA 3 excels in classifying "Severity" and "Root Cause." GPT-3.5, although lagging behind GPT-4, exhibits commendable accuracy. The insights gained from this study are vital for the construction industry, as they indicate the potential for developing more precise and effective safety measures. These findings could lead to a reduction in the frequency and severity of accidents, thereby enhancing worker safety.

1. Introduction

Maintaining high safety performance for construction companies has been challenging in many countries. Construction workers are often exposed to hazardous working environments on site that can lead to major injury or loss of life. According to reports from the International Labor Organization [1], more than 3 million deaths are caused by work-related accidents and diseases, with nearly 330,000 fatalities and work accidents. Additionally, the construction industry constitutes approximately one of six fatal accidents. Due to the high rate of accidents and fatalities in the construction sector, it is crucial to analyze previous incidents to prevent injuries and improve the safety of construction workers.

Understanding the root and injury causes of construction accidents is vital for preventing future occurrences. The "Fatal Four" - Falls, Struck-by, Caught in/between, and Electrocution - are the primary contributors to construction fatalities, accounting for a significant portion of accidents. For instance, between 2011 and 2020, over 25,000 fatal incidents in construction involved these causes, with Falls (35.1%), Struck by (17.1%), Caught in/between (5.8%), and Electrocution (7.7%) being the predominant ones [2]. Identifying these specific causes helps in targeting safety measures more effectively. Additionally, insights into the affected body parts, accident timing, and severity are critical for developing comprehensive risk management strategies. This data assists in crafting targeted safety protocols and emergency response plans, reducing the likelihood of severe injuries and fatalities.

Previous construction safety research has utilized various methodologies for accident analysis and classification. Studies have incorporated various machine learning and natural language processing techniques, including traditional classifiers like Support Vector Machines (SVM), decision trees, ensemble methods, and advanced deep learning approaches such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [3-8]. For example, Goh and Ubeynarayana 2017 [4] utilized SVM and other classifiers to effectively categorize construction accident narratives, achieving high precision and recall across different accident causes.

✉ Address correspondence to Chao Wang, chaowang@lsu.edu

Citation: Ehsan Ahmadi, Shashank Muley, Chao Wang. Automatic Construction Accident Report Analysis Using Large Language Models (LLMs). *J Intell Constr*, 2024, 2, 9180039.

Furthering the capabilities of text analysis, Zhang in 2022 [6] introduced a hybrid structured deep neural network that incorporates Word2Vec to enhance the semantic analysis of accident causes. This approach improved the classification's accuracy by leveraging deep learning to capture nuanced semantic relationships in accident data, demonstrating a significant advance over traditional models. Similarly, Luo et al. in 2023 [8] employed Convolutional Neural Networks (CNNs) specifically tailored for the construction industry, allowing for an advanced parsing and understanding of complex textual patterns in accident reports. This methodology enhanced the granularity with which accident causes were identified and analyzed, providing deeper insights into the contributory factors and mechanisms of accidents.

While these methods have been effective, they do not fully exploit the capabilities of modern Large Language Models (LLMs), which can process and understand large volumes of unstructured text data with greater contextual awareness. This represents a significant gap in the current research landscape, as LLMs offer the potential for deeper and more accurate insights into accident reports. Additionally, current studies often lack a comprehensive attribute analysis, which is essential for fully understanding construction accidents. A deeper exploration of crucial attributes, including accident causes, severity, affected body parts, and timing of accidents, can provide more targeted and effective safety measures.

In this study, we employ advanced Large Language Models (LLMs), such as GPT, Gemini, and LLaMA, to enhance the classification and understanding of key attributes in construction accident reports. These attributes include root cause, injury cause, affected body part, severity, and accident timing. This research introduces the pioneering application of state-of-the-art language models for detailed and nuanced analysis of construction accident reports, significantly improving the accuracy of classifying complex report attributes, which is critical for developing targeted and effective safety measures and providing a strong foundation for enhancing safety protocols in the construction industry, potentially reducing the frequency and severity of accidents.

2. Background

2.1. Evolution of Text Classification

In the realm of Natural Language Processing (NLP), text classification, akin to other NLP tasks, has experienced significant evolution. Earlier on, text classification depended on traditional machine learning models such as Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor, and Random Forest. These were often combined with text representation techniques like Bag-Of-Words, Word2Vec, and N-gram [9]. While effective for foundational classification tasks, these methods required extensive feature engineering and often struggled to capture the subtleties of language and its contextual nuances.

The emergence of neural network-based methods marked a notable advancement in text classification. Leveraging deep learning, these methods autonomously learn and extract textual features, thereby minimizing the need for manual feature engineering. Techniques such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Graph Convolutional Networks (GCNs) have been integral and successful in this domain [10-14]. However, these methods occasionally encounter challenges in fully grasping the entire scope of language context, particularly in interpreting complex sentence structures and diverse language usage.

In parallel, Large Language Models (LLMs) have risen as a pivotal development in NLP tasks, including text classification. Rooted in sophisticated architectures like transformers—a variant of neural networks—LLMs can process and comprehend extensive volumes of text with exceptional accuracy. Their architecture is particularly adept at tasks demanding deep contextual understanding and in-context learning [15], significantly advancing the potential for complex NLP applications. This advancement has allowed LLMs to surpass the previous limitations of earlier methods, demonstrating remarkable performance in various text classification scenarios [16-18].

2.2. Large Language Models (LLMs)

Large Language Models (LLMs) are a class of deep learning models designed to understand, generate, and sometimes translate human language. They are characterized by their immense size, containing hundreds of billions of parameters, and are trained on vast text corpora to perform a wide array of language tasks [19]. The architectures of LLMs can be broadly categorized into three types[18].

The first type is encoder-only models like BERT [20]. These models focus on understanding the context of a given text. They use a masked language model objective during pre-training, where random tokens are masked out, and the model learns to predict them, thus gaining a deep understanding of language context and word relationships.

Decoder-only models, such as GPT [21], form the second type. These models use a unidirectional approach where each token can only attend to previous tokens in the sequence. This architecture is particularly suitable for tasks that involve generating text, as the model is trained to predict the next token in a sequence based on the tokens that came before it.

The third type comprises encoder-decoder models like T5 [22]. Encoder-decoder models consist of an encoder component to understand the input text and a decoder component to generate the output text. This architecture allows for a comprehensive approach to text processing, encompassing understanding and generating capabilities in a unified framework.

LLMs have demonstrated various advanced capabilities that significantly enhance their utility in various applications. Notable examples include zero-shot learning, in-context learning, fine-tuning, and step-by-step reasoning [23-25].

Zero-shot learning is a key feature where LLMs, as zero-shot learners, can answer queries they have never explicitly encountered before. This capability allows them to respond to

user questions without requiring examples in the prompt [24].

In-context learning, introduced by the developers of GPT [15], enables LLMs to understand and respond to prompts using contextual cues within the text. This approach is bolstered by few-shot learning, allowing these models to adapt to new tasks with minimal examples quickly. Unlike traditional models requiring extensive training, few-shot learning equips LLMs, like GPT, to perform tasks effectively with limited training instances, streamlining their adaptability to diverse applications.

Fine-tuning is another approach where LLMs can be fine-tuned in various ways to enhance their performance on specific tasks. Transfer Learning involves fine-tuning pre-trained models with task-specific data [22]. Instruction-tuning allows models to perform new tasks simply by reading instructions describing the task [23].

Step-by-step reasoning, facilitated by strategies like chain-of-thought (CoT) prompting [26], enables LLMs to process tasks that require multiple logical steps. This is particularly useful in complex problem-solving scenarios, allowing LLMs to provide more coherent and logically structured outputs.

3. Methodology

Building on the foundational knowledge of LLMs outlined in the previous sections, we aim to apply these technologies within the domain of construction safety. Utilizing the advanced capabilities of LLMs—such as zero-shot learning, in-context learning, and their ability to interpret vast amounts of unstructured text data—we conduct a comprehensive analysis of construction accident reports. Our methodology is structured into four distinct stages, as depicted in Figure 1, designed to leverage these models to categorize and analyze key attributes from the reports effectively.

3.1. Data Preparation

The initial phase of our methodology involves the preparation of data, which is crucial for ensuring the integrity and consistency of the input for model training and testing. We extract accident reports from the Occupational Safety and Health Administration (OSHA) database [27] spanning from

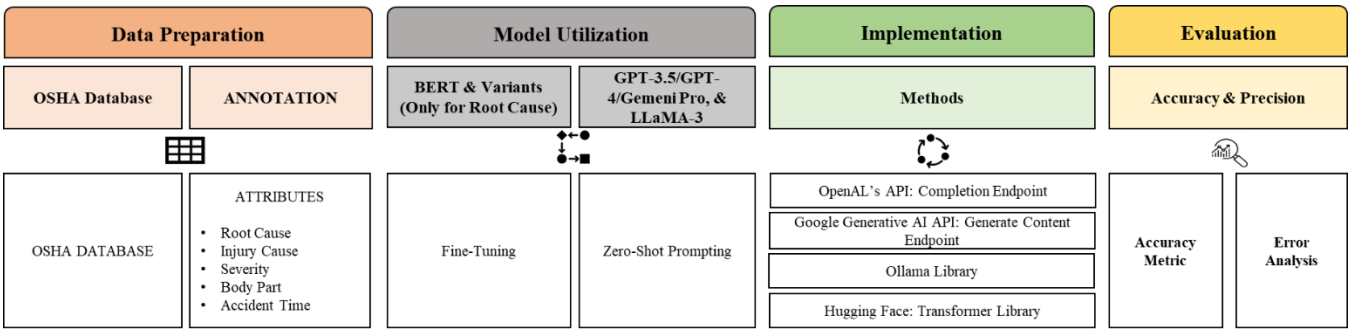

Fig. 1 The four-stage methodology framework encompassing Data Preparation, Model Utilization, Implementation, and Evaluation.

2002 to 2023. This comprehensive dataset allows us to cover a wide range of incidents, reflecting the evolving nature of workplace hazards and safety measures over two decades.

For fine-tuning BERT and its variants, we meticulously curated a subset of 1,000 reports specifically for root cause classification. This focus was chosen because fine-tuning LLMs requires a substantial amount of data, and root cause provides a clear baseline for classifying the underlying reasons behind incidents, facilitating direct comparison with more advanced models such as GPT, Gemini, and LLaMA. Furthermore, we designated an additional set of 185 reports to test all models' performance. This test set was carefully chosen to ensure the representation of all classes within each predefined attribute, with an average of 50 reports per class. Special consideration was given to reports detailing sequential events, particularly those elucidating the connection between root cause and injury cause, to enrich the model's understanding of causality in workplace accidents.

Attribute	Details
Report	At 1:00 p.m. on January 25, 2017, an employee was working in a suspended ceiling grid installing fire sprinkler piping. The employee came into contact with live electrical wiring and was pulled off a ladder by his Foreman, resulting in both falling to the floor. The employee injured his shoulder on impact with the concrete floor.
Injury Cause	Fall
Root Cause	Electrocution
Body Part	Shoulder
Severity	Non-fatal
Accident Time	1:00 p.m.

Table 1 Example Annotation of an Accident Report

Our annotation process, conducted manually by the authors, involved detailing each report according to the following predefined attributes, which are crucial for understanding construction accidents and commonly reported in accident documentation:

- Root Cause: Identifies the fundamental reason for the accident occurrence, categorized into types such as caught in/between, electrocution, falls, and struck-by incidents. This helps in understanding underlying factors for long-term safety improvements.

- Injury Cause: Specifies the direct cause of the injury, which may or may not be directly related to the root cause. It is essential for targeting specific safety interventions.
- Body Part: Indicates the body part affected, such as head, hand, leg, back, finger, torso, arm, etc. This information is significant for assessing the impact of accidents and tailoring protective measures.
- Severity: Classifies the accident's outcome in terms of severity, ranging from non-fatal to fatal, helping prioritize safety measures and resource allocation.
- Accident Time: Records the time at which the accident occurred, providing temporal context useful for identifying high-risk periods and planning safety operations.

An example of how these attributes are applied is provided in Table 1, which presents the annotation of a specific accident report. This detailed annotation approach allows for the comprehensive classification and analysis of accident reports, which is essential for our study's accurate application of LLMs.

3.2. Model Utilization

This phase involves the application of the following LLMs: BERT and its variants, GPT-3.5, GPT-4, Gemini Pro, and LLaMA.

- BERT & Variants: As foundational encoder-only models, BERT and its derivatives such as RoBERTa and DeBERTa enhance the interpretative depth by processing words in context to all other words within a sentence. These models are fine-tuned on our dataset specifically for the targeted approach of identifying the root cause of accidents. This focus is necessitated by the substantial data requirements for effective fine-tuning and provides a clear baseline for performance comparison across models.

- GPT-3.5, GPT-4, Gemini Pro, & LLaMA 3: The GPT series [28], including GPT-3.5 and GPT-4, are state-of-the-art decoder-only models that utilize layers of transformer blocks with multi-head self-attention mechanisms and feedforward neural networks, designed for generating text by predicting the next word in sequences. Gemini [29], developed by Google's DeepMind, is known for its robust handling of complex datasets through its advanced decoder-only model employing multi-query attention. We selected Gemini Pro for our study due to its enhanced capabilities and efficiency in processing extensive datasets. Similarly, LLaMA 3 [30], introduced by Meta, is a cutting-edge decoder-only model that incorporates advanced reasoning capabilities and instruction fine-tuning. We selected the 70B parameter model of LLaMA 3 for its demonstrated improvements in performance. LLaMA 3 utilizes grouped query attention and extended token sequences, making it particularly adept at processing extensive and complex text data. In our study, a zero-shot learning approach is utilized for these models. Universal prompts were meticulously engineered to standardize the classification process across the different models. These prompts were crafted to be clear and concise, aligning precisely with the input expectations of each model to ensure optimal performance. This design process involved careful consideration of the models' capabilities in context understanding and text generation, which is critical for accurate classification. Detailed information about the specific prompts used for each attribute in our study, including their exact wording and structure, is provided in Table 2. This approach facilitates a direct and fair comparison of model performance across a consistent set of tasks.

Table 2 Classification Prompts Used in the Study

Attribute	Prompt
Injury Cause	Determine the Injury Cause of the accident in the report. Your answer should be strictly one of the following: 'Electrocution', 'Struck by', 'Fall', or 'Caught in/between' without any additional text or explanations.
Root Cause	Determine the Root Cause of the accident in the report. Your answer should be strictly one of the following: "Struck by," "Caught in/between," Fall," Electrocution,' or 'Unspecified' without any additional text and explanations.
Body Part	Determine the Severity of the incident in the report. Your answer should be strictly one of the following: 'Fatal' or 'Nonfatal' without any additional text or explanations.
Severity	Determine the main Body Part affected in the accident. Provide only and strictly the main body part affected without any additional text or explanations. If the information is not available, say 'Unspecified'.
Accident Time	Determine the Accident Time of the accident in the report. The answer should strictly be in the format HH:MM am/pm. If the information is not available, say 'Unspecified'. Do not include the date and any other additional text and explanations.

3.3. Implementation

The implementation utilizes Python for all computational tasks. We use specific APIs for each model with tailored hyperparameters to optimize their performance for the interaction with the LLMs.

The GPT-3.5 and GPT-4 models are accessed via the OpenAI API's Completion endpoint. The settings for these models include using "get-3.5-turbo" and "get-4" as the Model Names, with a Temperature of 1, Top P of 1, and both Frequency Penalty and Presence Penalty set to 0.0. The 'Best Of' parameter is set to 1. This configuration aims to harness the GPT models' generative capabilities while ensuring output diversity.

In the case of the Gemini Pro model, our approach involved

interfacing with the Google Generative AI API, specifically employing the Generate Content endpoint. The configuration for the Gemini Pro model ("gemini-pro") involves a Temperature setting of 0.9, and Top P and Top K are set to 1. Additionally, safety settings are implemented to block content categories like harassment, hate speech, sexually explicit, and dangerous content at medium and above thresholds, ensuring responsible use of the model.

For LLaMA 3, we utilized the Ollama Library [31], interfacing with a locally hosted model set to a 70B parameter scale. The model configuration includes a Temperature of 1 and Top P of 1, optimized for generating precise and contextually relevant outputs.

In addition to these LLMs, the Hugging Face's Transformer Library [32] is employed to operationalize BERT and its variants. The BERT models are fine-tuned with a Learning Rate of 1e-5, employing 12 Attention Heads, a Hidden Size of 768, and an Embedding Size of 512. This setup aligns with standard BERT model configurations, making them suitable for diverse text classification tasks.

3.4 Validation and Evaluation

The validation of the models is performed using the manually annotated dataset detailed in the Data Preparation section, serving as the ground truth for verification. This rigorous evaluation process aims to quantify model accuracy comprehensively and explore the specifics of any misclassifications through detailed error analysis.

- Accuracy Metric: The primary measure for assessing the models' prediction accuracy against the annotated dataset. It is calculated as the ratio of correctly predicted instances (True Positives and True Negatives) to the total number of cases. The formula for accuracy is given by:

Equation (1)

$$Accuracy = \frac{TP + TN}{Total}$$

TP represents True Positives, TN represents True Negatives, and Total is the total number of instances.

- Error Analysis: This part of the study involves a detailed examination of the instances where the models misclassified certain attributes in the accident reports. By analyzing these errors, we gain insights into each model's specific challenges and limitations in accurately interpreting the complex narratives of construction accident reports.

4. Results and Discussion

4.1. Accuracy Results

Our analysis revealed varied performances across the LLMs utilized (Table 3). GPT-4 consistently outperformed GPT-3.5, demonstrating superior accuracy in classifying injury cause, root cause, severity, body part affected, and accident time. Remarkably, GPT-4 achieved perfect accuracy in determining the accident time, with a score of 100%. LLaMA 3 also achieved a perfect score of 100% in accident time classification, along with high scores of 95.65% in root cause and 99.46% in severity, closely aligning with the performance of GPT-4 in these attributes. The Gemini Pro model, while excelling in classifying the injury cause with an accuracy of 96.74%, demonstrated a marked disparity in accident time classification, scoring only 17.93%. LLaMA 3 also showed a strong performance with 96.20% accuracy in the injury cause classification, closely matching Gemini Pro. BERT and its variants showed consistent results, particularly in root cause classification, with scores in the mid-80s range, suggesting their less effective capabilities in this classification task.

Table 3 Accuracy Results

Attribute	GPT-3.5 (%)	GPT-4 (%)	Gemini Pro (%)	LLaMA3-70b (%)	BRET, RoBERT, DeBERTa(%)
Injury Cause	91.85	94.02	96.74	96.20	
Root Cause	94.57	97.83	89.67	95.65	83.30, 83.80, 83.62
Body Part	79.89	99.46	88.04	66.30	
Severity	88.04	94.57	86.96	99.46	
Accident Time	94.57	100.0	17.93	100.00	

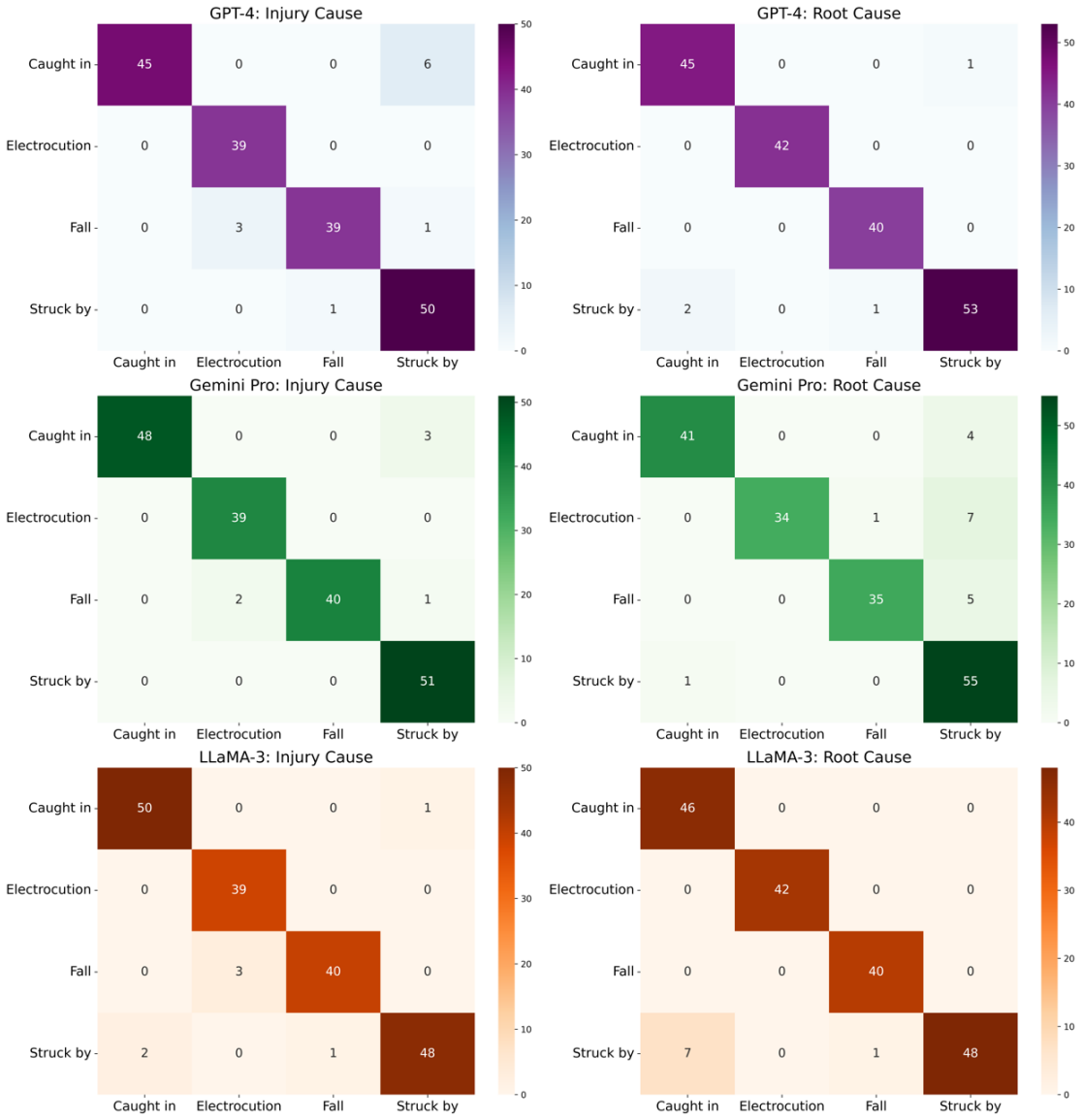

Fig. 2 Confusion matrices for injury cause and root cause classification using GPT-4, Gemini Pro, and LLaMA 3, with predicted classes on the horizontal axis and ground-truth classes on the vertical axis

4.2. Error Analysis

4.2.1. Injury Cause and Root Cause

Table 3 shows that GPT-4, Gemini Pro, and LLaMA 3 exhibited strong performance in classifying "Injury Cause," with each model demonstrating high accuracy. GPT-4, in particular, also stood out in accurately classifying "Root Cause," closely followed by LLaMA 3, which showed minimal misclassifications and effectively handled nuanced textual data. Analyzing the confusion matrices for these classifications (Figure 2), we observe distinct misclassification patterns from GPT-4, Gemini Pro, and LLaMA 3, revealing their interpretative strategies.

In the "Injury Cause" classification task, both GPT-4 and Gemini Pro have shown a pattern of misclassification where "Caught in/between" incidents are incorrectly classified as "Struck by." LLaMA 3 similarly misclassified one report as “Struck by” instead of the correct “Caught in/between,” although it more frequently made the opposite error, misclassifying “Struck by” incidents as “Caught in/between.” The common error of mislabeling "Caught in/between" as "Struck by," especially in GPT-4 and Gemini Pro, as depicted in the confusion matrices, might be due to the models' overemphasis on the verbs and agents suggesting motion or impact, which are prevalent in descriptions of "Struck by" events. For instance, in an accident where an employee is pinned against a trailer by a moving vehicle, both models are prone to categorizing the incident as "Struck by." This misclassification could arise from the models prioritizing the active dynamics of the incident—the hitting or colliding—over the passive but more accurate state of being "Caught in/between."

Furthermore, a similar pattern of misclassification by GPT-4,

Gemini Pro, and LLaMA 3 is observed, with "Fall" incidents being classified as "Electrocution." This points to an over-sensitivity to the context in which electrical elements are mentioned, even when they are not directly implicated in the cause of the fall. The models seem to be swayed by the presence of such terms in the text, leading to a misidentification of the cause as "Electrocution." This is evident from the confusion matrices where cases that should have been classified under "Fall" are instead categorized under "Electrocution" due to the models' potential misinterpretation of the electrical context as the dominant factor rather than as a secondary or unrelated aspect of the accident. This suggests that new strategies are required to better distinguish between the direct causes of accidents in such complex reports.

The "Root Cause" attribute analysis reveals divergent GPT-4, Gemini Pro, and LLaMA 3 performances. GPT-4 shows a relatively strong capability in identifying "Root Cause" with few misclassifications. However, it exhibits occasional confusion, such as categorizing a "Struck by" incident as "Caught in/between" and vice versa, suggesting a need for more nuanced differentiation within the model's decision-making process. On the other hand, Gemini Pro struggles significantly in this area, with a tendency to misclassify incidents like "Electrocution," "Fall," and "Caught in/between" as "Struck by." These errors suggest that Gemini Pro, more so than GPT-4, might assign undue importance to certain narrative elements that are not definitive of the "Struck by" category. Similarly, LLaMA 3 demonstrates a pattern where it favors "Caught in/between" classifications in scenarios involving both "Struck by" and "Caught in/between" dynamics, a tendency also observed in the "Injury Cause" classification. This indicates a potential model bias toward interpreting these incidents as more passive than they might be, particularly in complex accident scenarios with multiple dynamics. The pattern of misclassifications for Gemini Pro and the observed tendencies of LLaMA 3 implies a potential systemic challenge in discerning the initiating action from the resultant state in complex accident scenarios. Addressing these discrepancies may involve revisiting the classification approach or the prompt structure for Gemini Pro and LLaMA 3 to improve their understanding of the causal sequences commonly reported in construction accidents.

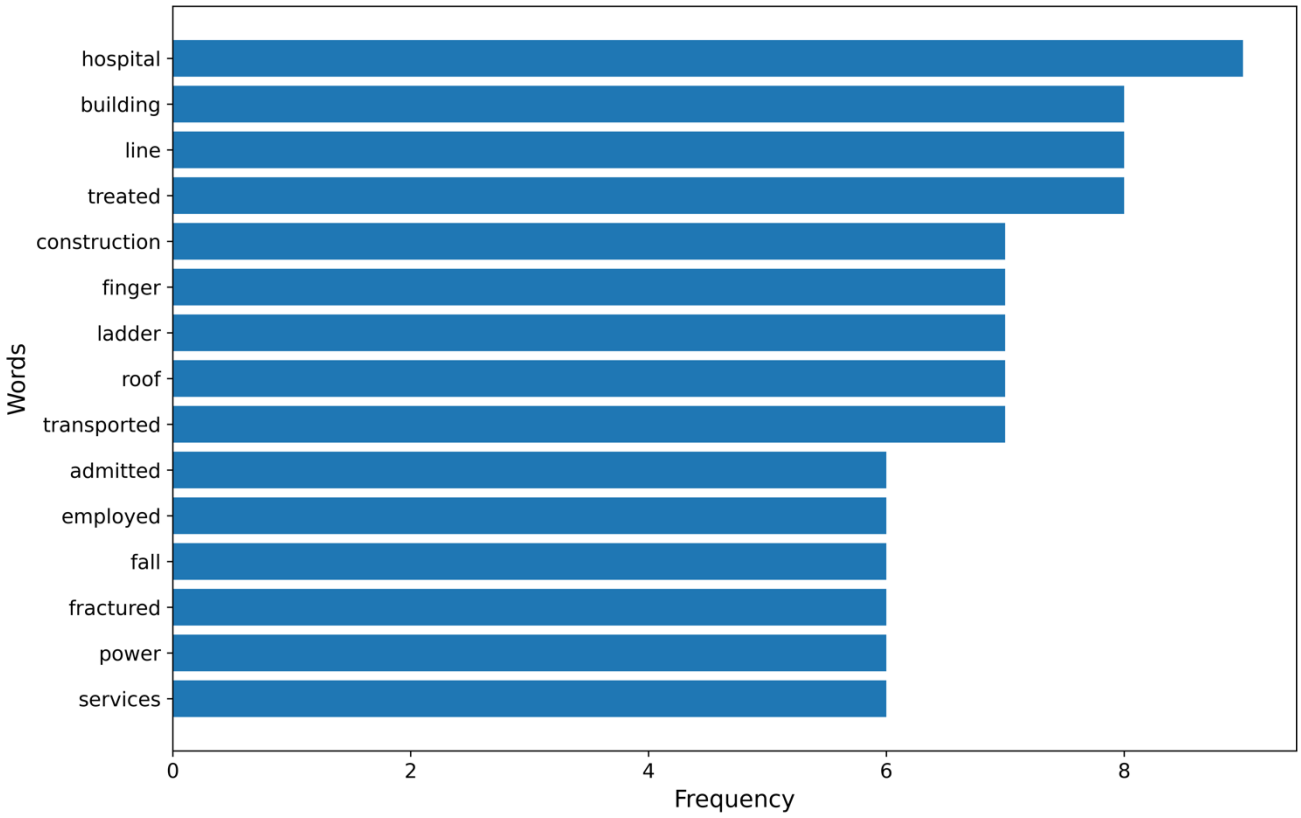

Fig. 3 Top words in reports incorrectly classified as "Fatal" by GPT-3.5 and Gemini Pro (more than 5 occurrences)

The confusion matrices highlight areas where sophisticated AI models like GPT-4, Gemini Pro, and LLaMA 3 may struggle with the complex nature of construction accident reports, especially when analyzing accident causes. The misclassifications indicate an opportunity to refine these models further to capture the nuances of sequential and layered events more effectively. While the current methodology is a solid foundation, enhancements in the models' interpretation of contextual cues through more detailed prompts or targeted fine-tuning could improve performance in future analyses.

4.2.2. GPT-3.5 vs. GPT-4 vs. Gemini Pro vs. LLaMA 3: Performance Discrepancies

The performance discrepancies among GPT-3.5, GPT-4, Gemini Pro, and LLaMA 3, as detailed in Table 3, illustrate distinct misclassification patterns across various attributes.

For Gemini Pro, the notably lower accuracy in the "Accident Time" classification might be attributed to a conservative strategy in interpreting the strict time format (HH: MM am/pm) specified in the prompt (see Table 2). The model possibly defaults to "Unspecified" if not confident in extracting a precise time, a potential reason for its high rate of unspecified responses.

In the classification of "Severity", GPT-3.5 and Gemini Pro exhibited a marked tendency to overestimate the severity of accidents. As demonstrated in Figure 3, a discernible pattern emerges from the frequency analysis of terms within misclassified reports. Despite the occurrence of the word "treated" in most of these reports, which could suggest non-fatal outcomes, both models exhibited an inclination to predict outcomes as "Fatal". This misclassification bias was particularly pronounced in hospitalization incidents, with "hospital" being a predominant term. The models seem to overvalue the implications of hospitalization, likely due to its association with severe injuries, as suggested by the commonality of words such as "transferred" and "fall" in the misclassified reports. Such terms may have led the models to infer a critical level of injury, thereby defaulting to the most severe "Fatal" category despite the actual non-fatal nature of the events. Conversely, GPT-4 and LLaMA 3 performed much better in accurately classifying severity, with LLaMA 3 achieving an impressive accuracy of 99.46%. This high level of accuracy indicates that LLaMA 3 effectively avoids the overestimation bias observed in GPT-3.5 and Gemini Pro, demonstrating a superior ability to interpret contextual cues and differentiate between fatal and non-fatal outcomes more accurately.

In the classification of "Body Part" injured, significant discrepancies were observed between the models GPT-3.5, Gemini Pro, GPT-4, and LLaMA 3, each revealing distinct capabilities in interpreting incomplete injury descriptions. Notably, GPT-4 frequently and correctly inferred the involved body part in scenarios where it was not explicitly mentioned, such as an asphyxiation incident where GPT-4 accurately identified "Chest" as the injured body part—a detail particularly missed by Gemini Pro, and less frequently by GPT-3.5 and LLaMA 3, which did manage to identify "Chest" correctly in some similar cases. This illustrates GPT-4's advanced contextual inference capabilities. In contrast, GPT-3.5 and LLaMA 3 sometimes incorrectly inferred body parts, with GPT-3.5 associating "Head" with falls and "Torso/Trunk" with caught-in/between accidents. LLaMA 3 also exhibited a tendency to infer specific body parts in such unspecified scenarios: predicting "Torso" in caught in/between accidents, "Heart" in electrocution cases, and "Head" in falls. This reflects a tendency to over-generalize based on the accident type rather than the specific details provided. Gemini Pro often defaulted to "Unspecified," avoiding potentially incorrect specific predictions but at the cost of valuable diagnostic detail. For example, in a case where an employee was caught between an I-beam and a safety bar on a lift, leading to mechanical asphyxiation, GPT-4, GPT-3.5, and LLaMA 3 correctly identified the "Chest," while Gemini Pro did not specify any body part. Additionally, when incidents explicitly involved multiple body parts, GPT-3.5 and GPT-4 were able to correctly identify these complex scenarios, whereas Gemini Pro continued to classify these as "Unspecified," underscoring its cautious yet often under-informative approach that contrasts sharply with the more assertive predictive models of the other three.

These variations in model performance highlight the intricacies of utilizing LLMs for text classification in the safety domain. While GPT-4 demonstrates proficiency in context understanding, Gemini Pro, LLaMA, and GPT-3.5 bring their unique strengths and face distinct challenges. The analysis underscores the importance of model choice and configuration in achieving accurate classification results.

5. Conclusion

Our research utilizing LLMs, notably GPT-3.5, GPT-4, Gemini Pro, and LLaMA 3, has significantly enhanced the analysis and classification of construction accident reports. This study focused on key attributes such as root cause, injury cause, affected body part, severity, and accident time. The results demonstrate that these LLMs, particularly GPT-4, have achieved high accuracy across most attributes. GPT-4 consistently outperformed GPT-3.5, while Gemini Pro excelled in classifying "Injury Cause." LLaMA 3 distinguished itself by accurately classifying “Severity” and “Root Cause.” These findings hold great promise for the construction industry, indicating the potential to develop more precise and effective safety measures, which could reduce the frequency and severity of accidents.

However, it is important to acknowledge the limitations inherent in our approach. While effective for most attributes, the zero-shot learning approach may not always capture the complex causal relationships as effectively as domain-specific models trained directly on construction accident data. Additionally, the study focuses primarily on the "Fatal Four" causes of construction accidents, potentially overlooking other less common but impactful causes. Addressing these broader categories in future research could provide a more comprehensive understanding of accident causes.

Future research should also explore the broader application of LLMs in construction safety, especially in predictive analytics. This expansion could further revolutionize safety management practices and contribute to creating safer work environments.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2222881. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Author contribution statement

All authors have given approval to the final version of the manuscript.

References

1. Organization, I.L. *Nearly 3 million people die of work-related accidents and diseases*. 2023 [cited 2023 11/12/2023]; Available from: https://www.ilo.org/global/about-the-ilo/newsroom/news/WCMS_902220/lang--en/index.htm.
2. CPWR. *The center for construction research and training - construction focus four*. 2023; Available from: https://www.cpwr.com/research/data-center/datadashboards/construction-focus-four-dashboard.
3. Tixier, A.J.P., et al., *Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports.* Automation in Construction, 2016. **62**: p. 45-56 DOI: https://doi.org/10.1016/j.autcon.2015.11.001.
4. Goh, Y.M. and C.U. Ubeynarayana, *Construction accident narrative classification: An evaluation of text mining techniques.* Accident Analysis & Prevention, 2017. **108**: p. 122-130 DOI: https://doi.org/10.1016/j.aap.2017.08.026.
5. Cheng, M.-Y., D. Kusoemo, and R.A. Gosno, *Text mining-based construction site accident classification using hybrid supervised machine learning.* Automation in Construction, 2020. **118**: p. 103265 DOI: https://doi.org/10.1016/j.autcon.2020.103265.
6. Zhang, F., *A hybrid structured deep neural network with Word2Vec for construction accident causes classification.* International Journal of Construction Management, 2022. **22**(6): p. 1120-1140 DOI: https://doi.org/10.1080/15623599.2019.1683692.
7. Alkaissy, M., et al., *Enhancing construction safety: Machine learning-based classification of injury types.* Safety Science, 2023. **162**: p. 106102 DOI: https://doi.org/10.1016/j.ssci.2023.106102.
8. Luo, X., et al., *Convolutional Neural Network Algorithm–Based Novel Automatic Text Classification Framework for Construction Accident Reports.* Journal of Construction Engineering and Management, 2023. **149**(12): p. 04023128 DOI: doi:10.1061/JCEMD4.COENG-13523.
9. Kowsari, K., et al., *Text Classification Algorithms: A Survey.* Information, 2019. **10**(4): p. 150.
10. Pengfei Liu, X.Q., Xuanjing Huang, *Recurrent Neural Network for Text Classification with Multi-Task Learning.* arXiv:1605.05101, 2016 DOI: https://doi.org/10.48550/arXiv.1605.05101.
11. Irsoy, O. and C. Cardie, *Deep recursive neural networks for compositionality in language.* Advances in neural information processing systems, 2014. **27**.
12. Zhang, Y. and B. Wallace, *A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification.* arXiv preprint arXiv:1510.03820, 2015 DOI: https://doi.org/10.48550/arXiv.1510.03820.
13. Conneau, A., et al., *Very deep convolutional networks for text classification.* arXiv preprint arXiv:1606.01781, 2016.
14. Yao, L., C. Mao, and Y. Luo. *Graph convolutional networks for text classification*. in *Proceedings of the AAAI conference on artificial intelligence*. 2019. DOI: https://doi.org/10.1609/aaai.v33i01.33017370.
15. Brown, T., et al., *Language models are few-shot learners.* Advances in neural information processing systems, 2020. **33**: p. 1877-1901 DOI: https://doi.org/10.48550/arXiv.2005.14165.
16. Balkus, S.V. and D. Yan, *Improving short text classification with augmented data using GPT-3.* Natural Language Engineering, 2023: p. 1-30 DOI: 10.1017/S1351324923000438.
17. Han, X., et al., *PTR: Prompt Tuning with Rules for Text Classification.* AI Open, 2022. **3**: p. 182-192 DOI: https://doi.org/10.1016/j.aiopen.2022.11.003.
18. Sun, X., et al., *Text classification via large language models.* arXiv preprint arXiv:2305.08377, 2023 DOI: https://doi.org/10.48550/arXiv.2305.08377.
19. Shanahan, M., *Talking about Large Language Models.* Commun. ACM, 2024. **67**(2): p. 68–79 DOI: https://doi.org/10.1145/3624724.
20. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018 DOI: https://doi.org/10.48550/arXiv.1810.04805.
21. Radford, A., et al., *Language models are unsupervised multitask learners.* OpenAI blog, 2019. **1**(8): p. 9.
22. Raffel, C., et al., *Exploring the limits of transfer learning with a unified text-to-text transformer.* Journal of machine learning research, 2020. **21**(140): p. 1-67.
23. Wei, J., et al., *Emergent abilities of large language models.* arXiv preprint arXiv:2206.07682, 2022 DOI: https://doi.org/10.48550/arXiv.2206.07682.
24. Naveed, H., et al., *A comprehensive overview of large language models.* arXiv preprint arXiv:2307.06435, 2023 DOI: https://doi.org/10.48550/arXiv.2307.06435.
25. Zhao, W.X., et al., *A survey of large language models.* arXiv preprint arXiv:2303.18223, 2023 DOI: https://doi.org/10.48550/arXiv.2303.18223.
26. Wei, J., et al., *Chain-of-thought prompting elicits reasoning in large language models.* Advances in neural information processing systems, 2022. **35**: p. 24824-24837.
27. OSHA. *OSHA Accident Report*. 2023 12-1-2023]; Available from: https://www.osha.gov/ords/imis/accidentsearch.html,.
28. Achiam, J., et al., *Gpt-4 technical report.* arXiv preprint arXiv:2303.08774, 2023 DOI: https://doi.org/10.48550/arXiv.2303.08774.
29. Team, G., et al., *Gemini: a family of highly capable multimodal models.* arXiv preprint arXiv:2312.11805, 2023 DOI: https://doi.org/10.48550/arXiv.2312.11805.
30. *Meta, Llama 3*. 2024; Available from: https://llama.meta.com/llama3/.
31. *Ollama*. 2024; Available from: https://ollama.com.
32. *Hugging Face Transformers Documentation*. 2024; Available from: https://huggingface.co/docs/transformers/.

Ehsan Ahmadi is currently a Ph.D. student majoring in construction management, and meanwhile also working towards his dual degree in master's in computer science at the Louisiana State University. His research interests lie in construction automation, construction robotics, and large language models. Ehsan is investigating the worker intent recognition from speech commands for exoskeleton control to better assist the worker locomotion.

Shashank Muley is a Ph.D. candidate in construction management at Louisiana State University and recipient of the SEC Emerging Scholar 2023-2024 in recognition of his excellence in research and leadership. His research focuses on construction safety, human behavior, wearable sensors, and data analytics. With years of industry experience in both the industrial and commercial construction sectors, he is working towards improving worker safety by adopting advanced technologies at both individual and organizational levels.

Chao Wang, Ph.D. is currently an associate professor and the graduate program advisor in the Bert S. Turner Department of Construction Management at the Louisiana State University. Dr. Wang graduated from Georgia Tech in 2014 with a Ph.D. in Civil Engineering specializing in Construction Engineering and Management, and his main research interests lie in construction automation and robotics, worker safety and health, and facility energy efficiency. Dr. Wang has been awarded as PI/Co-PI over $20 million in research funding from federal agencies such as the National Science Foundation (NSF), the Environmental Protection Agency (EPA), the U.S. Department of Energy (DOE), United States Department of Agriculture (USDA), and the U.S. Department of Transportation (DOT), and some other state funding agencies and industry companies as well.