2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

LLM-BASED PROCESSING OF DESIGN INSPECTION REPORTS AS A MEASURE OF BUILDING DESIGN QUALITY

Hamada Elshaboury[1], Fulvio Re Cecconi[1], Vincenzo Scotti[2], Luciano Baresi[1], Enrico De Angelis[1]
[1]Politecnico di Milano, Italy
[2] Karlsruhe Institute of Technology (KIT), Germany

Abstract

Extracting shareable knowledge concerning design-related issues from construction project documentation enables the assessment of design quality and, more broadly, improvement across the industry. This study proposes a solution to automate information extraction from inspection reports in construction industry processes, with a focus on those produced by design reviews often conducted before the tendering phase. The approach is based on LLMs, prompt engineering, and few-shot learning. We evaluated three LLMs (GPT-4o, Mistral, and Llama 3) across four-shot scenarios, assessing their performance, computational cost, and time. Results show that GPT-4o achieved the highest performance while ranking second in computational cost and time.

Introduction

Although it is widely acknowledged that design is a critical phase, especially for complex projects, the Architecture, Engineering, and Construction (AEC) industry often fails to implement the simplest preventive actions that would enable clients, designers, contractors, and even users to learn from errors, particularly those related to design. Design documentation may be incomplete or inconsistent. In addition, it may describe projects that are not buildable (within the specified timeframes or expected costs), do not comply with objectives and constraints, or suffer from anticipated performance losses, thus requiring maintenance sooner than expected. Addressing these issues during the construction phase often requires design changes or rework, resulting in project delays, cost overruns, disputes among stakeholders (Tilley and McFallan, 2000), and limited utilization or retrofit works. This leads to low productivity rates, negatively affecting the whole construction industry, a sector considered a catalyst for economic growth in all countries (Buckley *et al.*, 2016).

Existing studies on design quality issues in the AEC sector are mainly based on interviews and questionnaires answered by experts from different perspectives (e.g., designers, owners, contractors, and sub-contractors). These studies have helped identify the most frequent quality issues (Tilley and McFallan, 2000), proposed indicators (Philips-Ryder *et al.*, 2013), and their rankings. Although this information is reasonable, it is often biased, lacks detail, and is not always valuable, making it difficult to derive actionable insights for improving design. On the other hand, a vast quantity of documents and detailed information is generated during the design and construction phases (e.g., inspection reports, change orders, commissioning notes, and claims). However, these documents are not easily exploited from one project to another, or even within the same project, to identify to-do and not-to-do actions.

Our objectives are twofold: first, to *extract reliable information* from a specific category of documents, design inspection reports, which are a potentially rich source of information currently disregarded and inaccessible due to confidentiality constraints. These documents have been produced in the validation process of every Italian public project since 1994, while similar review processes are commonly carried out in many other countries. Second, to *extract knowledge* from this information, promoting learning from errors as a practice in the AEC sector. To this end, we initiated research to investigate how this information can be processed and used to transfer prevention-related knowledge among stakeholders in the field. Artificial intelligence (AI) was also employed to avoid time-consuming and costly manual mining, as structured in the research framework illustrated in Figure 1.

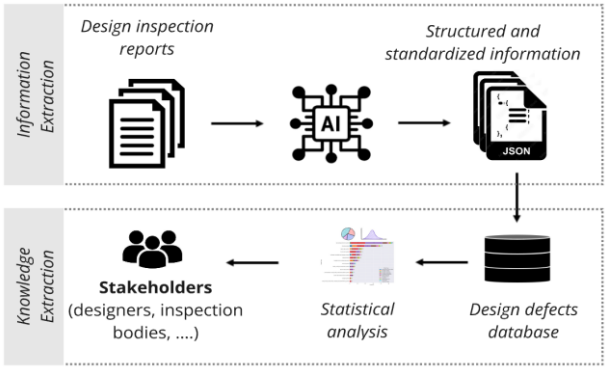


Figure 1: Research framework (steps)

In this paper, we address the information extraction step of the framework by developing a methodology in three steps: (1) data preparation, (2) identification of information entities, and (3) information extraction automation

using Natural Language Processing (NLP) and Large Language Models (LLMs).

This paper is structured as follows: an introductory State of the Art section, a Methodology section outlining the proposed approach, an Experiments section, and finally, the Conclusion and References.

State of the Art

Design documentation functions and quality

Design is a "*set of processes that transforms requirements for an object into more detailed requirements for that object*" (ISO 9000:2015). It serves three main aims: to inform stakeholders about design choices, demonstrate their validity against design input requirements, and specify the obligations the selected contractor is expected to fulfill. The design, as an activity, produces a model of the objects to be realized (in written, graphical, or digital formats) and a series of reports to demonstrate the validity of the design choices. Therefore, design quality (or conversely, design defects) is not only the ability of these documents *"to provide the contractor with all the information needed to enable construction to be carried out as required, efficiently and without hindrance*" (Tilley *et al.*, 1997), that is their completeness, consistency, and buildability. It is also reflected in the compliance of the realized object with its requirements, the accountability of the general description, and the clarity and correctness of the reports that demonstrate the feasibility of achieving the specified requirements.

Many research efforts have been conducted to investigate design document quality issues, indicators, attributes, and the causes of design defects, including the failures they may lead to and their relative importance. It is worth noting that most of these studies rely on expert opinions (e.g., questionnaires and interviews, which are not always the most reliable sources), offer a low level of detail, and leave a gap in eliciting actionable knowledge to enhance design quality.

NLP-based text processing

In the AEC sector, vast amounts of information are recorded in paper-based documents. Therefore, Natural Language Processing (NLP), a branch of Artificial Intelligence (AI) and linguistics, has emerged as a powerful approach to transforming unstructured textual data into structured information, thereby enhancing decision-making (Ding *et al.*, 2022). Numerous research efforts have explored various NLP-based methods for this process, including rule-based approaches, Machine Learning (ML), Deep Learning (DL), and, more recently, Large Language Models (LLMs). Rule-based methods utilize predefined patterns derived from document analysis and linguistic rules crafted by experts, which the model then applies to extract matching information from unstructured data. For example, (Xu *et al.*, 2021) proposed a rule-based NLP approach to extract information from Chinese subway accident reports. More recent ML-based methods learn from training data to identify patterns and apply them when processing unseen data. (Salama and El-Gohary, 2013) proposed an ontology and Support Vector Machines (SVM)-based approach to classify clauses and subclauses within general conditions of construction contracts.

DL-based methods excel in understanding complex texts by leveraging neural networks. (Moon *et al.*, 2022) developed a Bidirectional Long Short-Term Memory with a Conditional Random Field (Bi-LSTM-CRF)-based Named Entity Recognition (NER) approach to extract information from road construction specifications. However, a significant limitation of these approaches lies in the substantial manual effort required to create matching patterns for rules-based methods or annotate training data for ML and DL models.

Since the introduction of transformer architecture and the self-attention mechanism (Vaswani et al., 2017), pretrained large language models (LLMs) such as the GPT family by OpenAI (OpenAI *et al.*, 2023) and LLaMA by Meta AI (Touvron *et al.*, 2023) have significantly advanced the field of NLP. These models adopt a prompt-based paradigm, allowing them to perform tasks by interpreting contextual instructions provided in prompts. This evolution has given rise to prompt engineering and in-context learning techniques (Brown *et al.*, 2020). Recently, LLMs have been adopted in the construction sector. (He *et al.*, 2024) proposed a BERT and GPT-based approach to segment and classify constraints in onsite meeting transcripts. (Elshaboury *et al.*, 2024) developed a transformer-based NER and Relation Extraction (RE) model to extract information from design and construction-phase documents, supporting building management. (Tran *et al.*, 2024) integrated Mistral with a Retrieval-Augmented Generation (RAG) system to interpret user queries, retrieve relevant information, and generate responses, enhancing construction site safety. Similarly, (Smetana *et al.*, 2024) proposed a GPT-3.5-based model to extract, summarize, and analyze injury reports, contributing to improved safety in highway construction.

Methodology

This section presents the proposed methodology, structured into three steps, as illustrated in Figure 2. The following subsections provide a detailed discussion of each step.

Step 1: Data preparation

This step focuses on preparing the unstructured design inspection reports for the processing step. It includes both data collection and preprocessing, as outlined below:

- *Data collection*: we collected a corpus of design inspection reports from Italian construction projects with the support of one of the country's largest engineering companies based in Milan. Each report contains project metadata (e.g., name, location, area, bid price, work scope, year), inspection process details (e.g., inspection scope and verification checks), a list of design documents submitted by the designer, and identified design defect cases resulting from the application of predefined verification checks. These checks address defects such as inconsistencies, missing details, and inadequacies.

- *Data preprocessing*: we prepared the collected documents by applying a sequence of text-cleaning transformations, including removing non-text elements (e.g., figures, headers, and footers), deleting duplicate cases, dividing complex cases, and replacing document IDs with their corresponding names for better understanding and association. Finally, the preprocessed data was saved in plain text format for LLM processing.

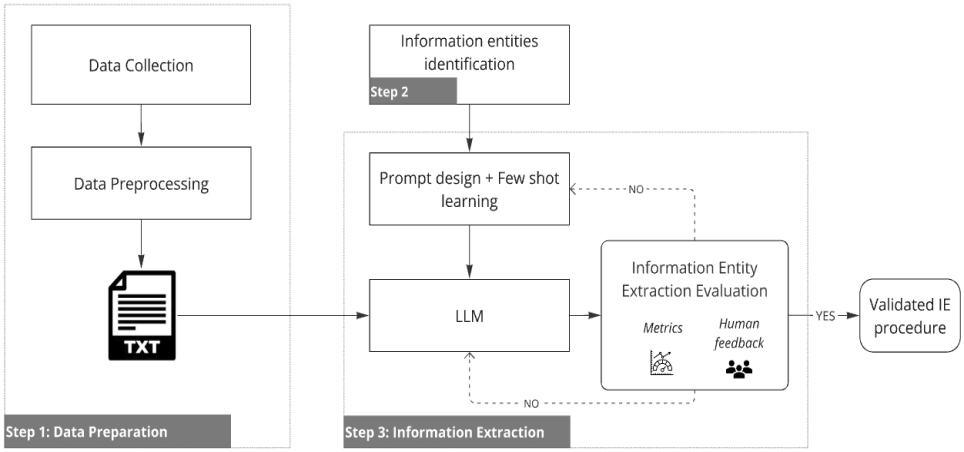


Figure 2: Methodology workflow for Information Extraction (IE) automation

Step 2: Identification of information entities

Identifying target information is fundamental to determining which entity categories should be extracted from unstructured texts describing design quality issues (i.e., defects). This process was guided by two key questions: (1) *Which* information is present in texts describing defects? and (2) *Why* is this information useful?

To answer the first question, we conducted a content analysis of design inspection reports to examine both the textual content and the typical structure used by design reviewers to formulate non-conformity cases. The analysis highlights that reviewers consistently mention document names, building elements, and the nature of the design defect, as illustrated in Figure 3. Based on these findings, four key information entity categories were identified as essential for effectively structuring defect information, as summarized in Table 1.

Addressing the second question (*why* this information is useful) helps clarify the purpose of extracting each entity and the type of knowledge derived. The "Checked Document" entity identifies which design documents are associated with specific defects, highlighting those with a higher frequency. The "Referenced Document" entity reveals relationships between design documents, enabling the identification of inconsistencies and quality issues. The "Defect" entity structures each reported issue, enabling faster comprehension and tracking while supporting analytical processes such as defect clustering. Lastly, the "Object" entity links defects to specific building elements, spaces, or systems, enabling designers and inspection bodies to recognize defects related to each designed component and systematically address them.

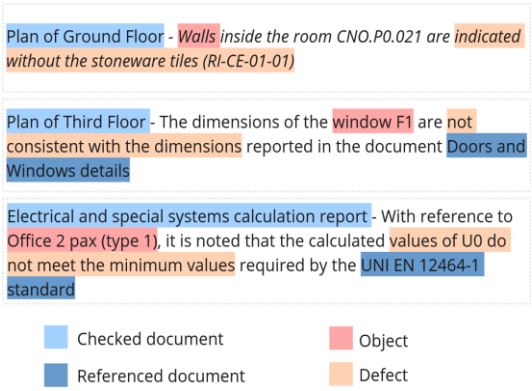


Figure 3: Examples of design defect texts from an inspection report

Table 1: Information entity Categories

Entity Category	Definition
Checked document	The name of the document(s) under inspection in which the defect is identified (e.g., structural calculation)
Referenced document	The name of the document(s) associated with the information inconsistency or non-compliance defect (e.g., regulations)
Object	The physical building element, system, or space related to the defect (e.g., wall, living room)
Defect	A summary of the described design-related issue pertaining to either an object or document (e.g., missing sections)

Step 3: LLM- based information extraction (IE)

We used Large Language Models (LLMs) as a means to extract the information of interest due to their advanced capabilities in comprehending and processing complex texts. Prompt engineering and In-context learning (i.e., few-shot learning) techniques were employed to guide the models in performing the extraction task without fine-tuning. This step includes three sub-steps, as detailed below:

Step 3.1: Prompt development

This subsection outlines the process of prompt development in four steps (Marvin *et al.*, 2024), (Jeon *et al.*, 2025) as depicted in Figure 4:

- *Define prompt schema*: Outlines the main components such as the persona of interest (LLM's expertise), task description (objective and required actions), task instructions (what to extract), output format (structure of the output), and additional notes (for guidance or clarification).
- *Develop task prompt:* Creates a detailed prompt tailored to the information extraction task, based on the predefined schema, to guide the model.
- *Evaluation*: Tests the developed prompt by processing sample inputs. The correctness and consistency of the model's responses are assessed through human feedback, leading to prompt validation or refinement.
- *Refine prompt*: Analyzes any errors or inconsistencies identified in the results and modifies the prompt by rewording, adding more examples, or providing further context.

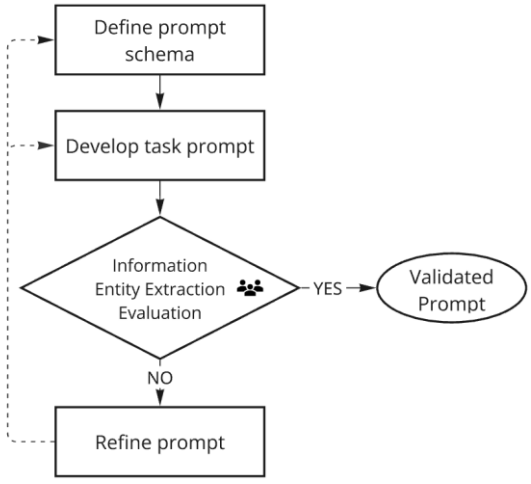


Figure 4: Prompt Development workflow

Step 3.2: Information extraction pipeline

Developing a robust pipeline for information extraction is essential to ensure a seamless and logical flow of information. As illustrated in Figure 5, the proposed pipeline comprises two main functions:

- *Function 1*: Handles the extraction of predefined information entity categories from each defect text. It establishes a connection with the LLM by preparing the input, which includes a system message (model persona and task description) to provide general task guidance, task instructions, and few-shot examples to provide contextual understanding, and finally, sending the input to the model and receiving its response.
- *Function 2*: Processes the list of defect texts by invoking *Function 1* for each defect text. Upon receiving the response from *Function 1*, it parses the returned content to extract the relevant information entity categories and appends them to a predefined results list. Once all defect texts have been processed, the aggregated information is returned in JSON format.

```
DEFINE Function 1 to extract information using LLM:
    GET text, few-shot examples
    ADD prompt to messages as system message and instructions
    ADD few-shot examples to messages
    ADD text to messages
    CALL LLM chat completion API and upload messages
    RETURN response content

DEFINE Function 2 to process defect texts list:
    GET defect texts list, few-shot examples
    SET results list
    FOR each text in defect_texts_list:
        CALL Function 1 with text and few-shot examples
        PARSE response content from LLM for information entities
        APPEND information entities to results list
        PRINT current processing status
    RETURN results_list

CALL Function 2 to start the extraction process
```

Figure 5: Pseudocode of IE pipeline

Step 3.3: Evaluation metrics

In this study, five metrics were used to assess the IE results robustly: Exact Match, BLEU, ROUGE-N, BERTScore, and SBERT. Exact Match is a binary metric that assigns a value of 1 if the predicted output exactly matches the ground truth and 0 otherwise. BLEU (Papineni *et al.*, 2002) and ROUGE-N (Lin, 2004) are n-gram-based metrics measuring word overlap between the predicted text and the ground truth. BLEU focuses on precision, evaluating how much of the predicted text accurately matches the reference, with a brevity penalty applied to short generated text, as simplified in Eq. 1. In contrast, ROUGE-N measures the F1 score, calculated as the harmonic mean of precision and recall, as shown in Eq. 2. Both BERTScore (Zhang *et al.*, 2019) and SBERT (Reimers and Gurevych, 2019) are semantic similarity metrics that utilize embeddings to measure the similarity between the predicted text (C) and the ground truth (R), as shown in Eq. 3. While BERTScore calculates word-level semantic similarity, making it more suitable for short texts, SBERT uses sentence-level embeddings, making it more effective for large-scale text comparison.

$$BLEU = \frac{overlap\ n{-}gram}{total\ n{-}gram\ in\ predicted\ text} \quad (1)$$

$$ROUGE-N = \frac{2\ (overlap\ n{-}gram)}{total\ n{-}gram\ in\ ground\ truth\ and\ predicted\ text} \quad (2)$$

$$BERTScore = \frac{1}{N}\sum_{i=1}^{N} max\ (cos\ (C_i, R)) \quad (3)$$

Experiments

In this section, the proposed methodology is evaluated using a real-world dataset. This process comprises data preparation, implementation, and results evaluation.

Data preparation

A set of design defect texts was collected from design inspection reports to create the reference dataset (i.e., ground truth) for LLM evaluation and few-shot learning preparation. Following the procedure described in Step 1, a reference dataset was developed, comprising 276 design defect cases. In each case, the predefined information entity categories listed in Table 1 were manually extracted and structured in an Excel file. From the reference dataset, 15 cases were selected based on defect types to prepare the few-shot learning scenarios, while the remaining 261 cases were used for the evaluation process.

Implementation

To process the design defect texts following the proposed methodology, three LLMs were employed: GPT-4o from (Open AI), Mistral Large from (Mistral AI), and Llama 3 from (Meta AI). All models were accessed via a cloud-based API. The temperature parameter was set to 0 to ensure consistent responses, promoting deterministic output and minimizing variation in the results. For few-shot learning, four scenarios were used: zero-shot, five-shot, ten-shot, and fifteen-shot, to evaluate the models' performance. The examples for each scenario were prepared in JSON format. For evaluation, the predefined metrics were executed using the *Evaluate* library from Hugging Face.

Evaluation results and discussion

The models were evaluated from three perspectives: performance, computational cost, and time, as detailed below.

Performance

Table 2 presents the average performance scores of the three models across the evaluation metrics and the four-shot scenarios. The results indicate that GPT-4o achieved the highest performance across all evaluation metrics and shot configurations. Specifically, GPT-4o outperformed Mistral with an average improvement of 11%, 7%, 4%, 5%, and 2% in Exact Match, BLEU, ROUGE-N, BERTScore, and SBERT, respectively. Compared to Llama, GPT-4o showed improvements of 13%, 16%, 8%, 9%, and 5% across the same metrics. Meanwhile, Mistral outperformed Llama by 2%, 8%, 3%, 4%, and 3%, respectively. Regarding few-shot learning, the results demonstrated that the performance of all three models improved as the number of shots increased, albeit at different rates. From 0 to 5 shots, GPT-4o improved by 9%, Mistral by 13%, and Llama 3 by 10%. Between 5 and 10 shots, GPT-4o showed minimal improvement (1.0%), while Mistral and Llama 3 increased by 3.5% and 4%, respectively. From 10 to 15 shots, GPT-4o and Llama 3 both improved by approximately 3.5%, whereas Mistral showed only 1.0%. Overall, Mistral achieved the highest improvement score in performance across the few-shot scenarios.

Comparing the values across different metrics, Exact Match yielded the lowest scores, indicating that achieving word-for-word alignment with the reference text was particularly challenging for specific information entity categories. For n-gram overlap metrics, BLEU and ROUGE-N scored higher than Exact Match but remained below the semantic similarity metrics. This suggests that while the models often captured keywords aligned with the reference data, they did not consistently produce identical matches. In contrast, BERTScore and SBERT achieved the highest values among all models, indicating that the extracted outputs were semantically close to the reference, even when vocabulary or syntax differed.

Table 2: Performance of LLMs for the IE Task

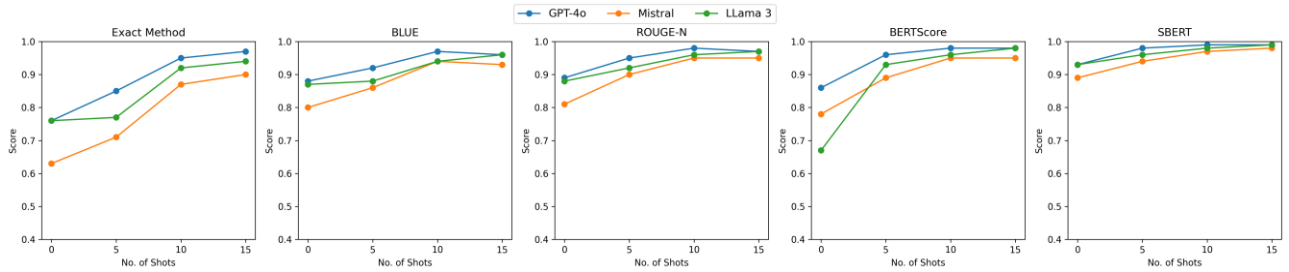
LLM	Evaluation Metric	No. of shots			
		0	5	10	15
GPT-4o	Exact Match	0.54	0.59	0.61	0.64
	BLEU	0.61	0.66	0.67	0.71
	ROUGE-N	0.69	0.75	0.75	0.77
	BERTScore	0.71	0.80	0.80	0.82
	SBERT	0.82	0.87	0.87	0.89
Mistral	Exact Match	0.46	0.53	0.57	0.58
	BLEU	0.54	0.62	0.66	0.66
	ROUGE-N	0.65	0.72	0.73	0.74
	BERTScore	0.66	0.77	0.78	0.78
	SBERT	0.80	0.86	0.86	0.87
Llama 3	Exact Match	0.48	0.50	0.55	0.57
	BLEU	0.51	0.56	0.59	0.63
	ROUGE-N	0.64	0.69	0.70	0.72
	BERTScore	0.61	0.74	0.75	0.77
	SBERT	0.78	0.83	0.84	0.85

Regarding the performance of LLMs in each information entity category, Figure 6.a shows that GPT-4o and Llama 3 achieved good performance for entity "*Checked document*", with scores above 0.90 across all metrics from 10 shots onward. At 15 shots, performance across all models remained stable or slightly declined, except for Exact Match, suggesting that 10 shots may be the optimal threshold for extracting this entity.

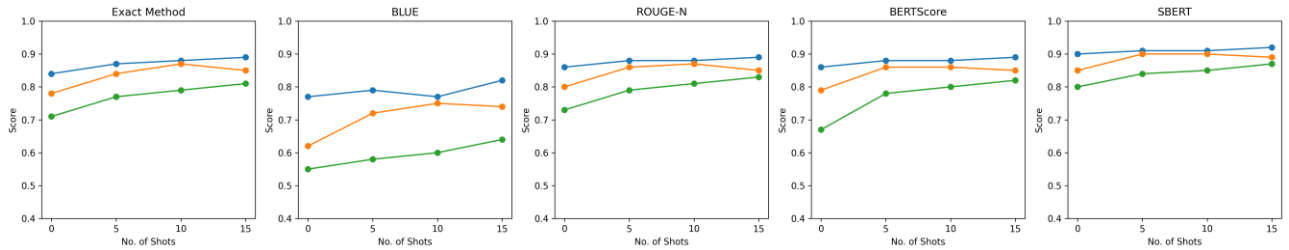
For the "*Referenced document*" entity, Figure 6.b shows that GPT-4o and Mistral achieved strong performance, with scores ranging from 0.80 to 0.90 at both 5 and 10 shots across all metrics except BLEU. Additionally, both models exhibited performance stabilization or a slight decline at 15 shots, suggesting that 5 or 10 shots may be the optimal setting for accurate extraction.

For the "*Object*" entity, Figure 6.c shows that GPT-4o and Mistral followed a similar performance trend. Their scores ranged from 0.70 to 0.75 at 5 and 15 shots according to the ROUGE-N and BERTScore metrics and exceeded 0.8 based on the SBERT metric. The results from Exact Match and BLEU suggest that these metrics are less suitable for evaluating model performance for this entity.

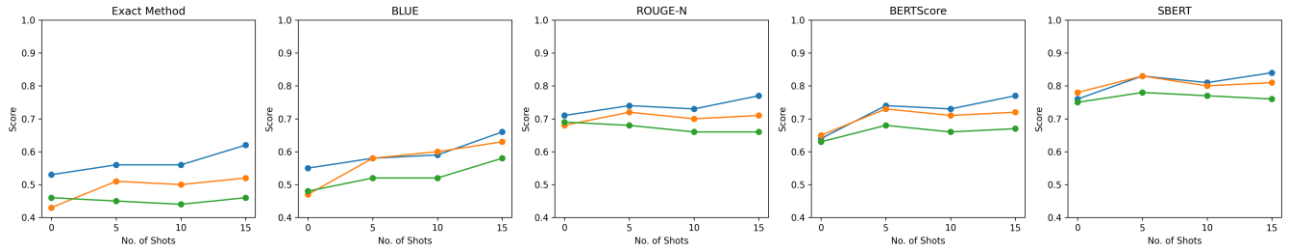
For the "*Defect*" entity, as shown in Figure 6.d, the three models exhibited a similar performance trend, with scores closely aligned within each metric. BERTScore and SBERT indicate that each model maintained consistent performance from 5 shots onward, ranging from 0.57 to 0.62 for BERTScore and 0.75 to 0.80 for SBERT. In contrast, Exact Match, BLEU, and ROUGE-N demonstrated limited effectiveness, reflecting the inherent nature of the "Defect" entity. Since the models were required to summarize the defect text, word order and syntax variations occurred, making these metrics less suitable for evaluation. Instead, semantic similarity metrics (i.e., BERTScore and SBERT) provide more reliable insights into extraction quality for this entity.



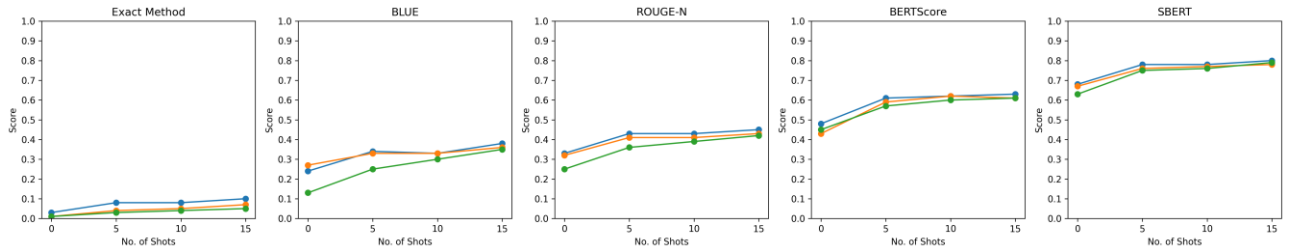
a) "Checked document"



b) "Reference document



c) "Object"



d) "Defect"

Figure 6: Performance of LLMs in Extracting Information Entity Categories

Computational cost

The computational cost is primarily influenced by the input text length and the generated output, measured in tokens as defined by the model's tokenizer. As illustrated in Figure 7, the number of input tokens increased with the number of shots, with Llama 3 producing the highest token count, followed by Mistral and GPT-4o. Conversely, the number of output tokens decreased as the shot count increased, suggesting that models generated more concise responses aligned with the ground truth when provided with additional contextual examples. Regarding cost, Llama 3 incurred the highest computational expense, followed by GPT-4o, while Mistral remained the most economical. Although GPT-4o utilized the fewest tokens, its higher per-token pricing than Mistral resulted in moderate computational expenses.

Computational time

As illustrated in Figure 8, Mistral achieved the lowest median response time across most shot counts, indicating a high efficiency in handling increased in-context input. GPT-4o recorded the lowest median time at zero shots and showed competitive time at five shots, but its response times were higher than those of Llama 3 at 10 and 15 shots. Regarding the Interquartile Range (IQR), GPT-4o demonstrated the most consistent performance, followed by Mistral, while Llama 3 exhibited the broadest IQR across all shot counts, indicating greater variability. In terms of outliers, Mistral displayed the lowest number across all shot counts, reflecting high-performance stability, followed by GPT-4o and Llama 3. Overall, Mistral Large achieved a well-balanced combination of efficiency, performance stability, and consistency.

Error analysis

Error analysis was conducted on the results produced by GPT-4o after 15 shots to investigate the sources of errors and identify opportunities to improve performance. Two main errors were identified. First, the model exhibited extraction format errors when processing texts that contained multiple document names, objects, or defects. For example, the model extracted *"doors of the WC cubicles and shower cubicles"* as a single object instead of *"doors of the WC cubicles, doors of the shower cubicles"*. This distinction is significant for downstream classification and statistical analysis. Therefore, more explicit instructions will be added to the prompt to generate more consistent and structured outputs, such as: *<Separate all extracted document names, objects, defects by comma>*, *<If the object includes more than one item with the same prefix, repeat the prefix for each one>*. Second, the model showed extraction errors for the "Defect" and "Object" information entities in complex cases, notably when the few-shot examples did not include similar instances to the input text. This suggests that increasing the number of shots is required to improve model performance, although this may be an ineffective solution from a cost perspective. To address this challenge effectively from both performance and cost perspectives, future work will incorporate Retrieval-Augmented Generation (RAG) to support the methodology through a dynamic few-shot approach.

Conclusions and future work

This paper proposed an LLM-based methodology integrating prompt engineering and few-shot learning techniques for Information Extraction (IE) automation from design inspection reports, aiming to support improvements in building design quality. The study identified and utilized four information entity categories to effectively structure and extract relevant design defect information. The methodology was experimentally evaluated using three LLMs (GPT-4o, Mistral, and Llama 3) across four shot scenarios, assessing performance, computational cost, and time. GPT-4o ranked first in performance at fifteen shots, achieving Exact Match (0.64), BLEU (0.71), ROUGE-N (0.77), BERTScore (0.82), and SBERT (0.89), while ranking second in computational cost and time after Mistral. Based on the results: (1) GPT-4o will

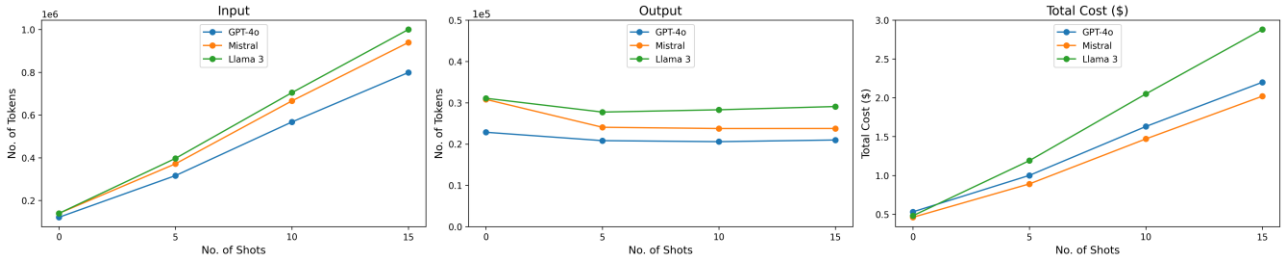


Figure 7: Computational cost of LLMs

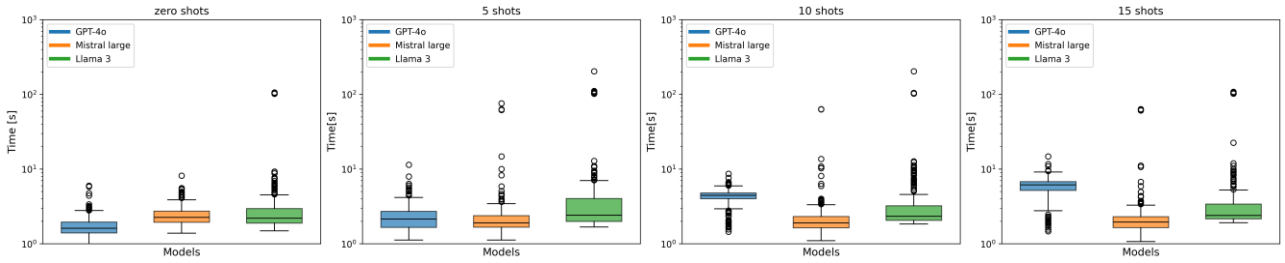


Figure 8: Computational time of LLMs

be adopted as the base Information Extraction (IE) model to process additional design inspection reports, and (2) semantic similarity metrics (i.e., BERTScore and SBERT) will be used to evaluate model performance for extracting the "Object" and "Defect" entities, as they provide more reliable insights into extraction quality. In addition, based on the error analysis, two modifications to the proposed methodology are planned: (1) a revised version of the prompt will be developed with improved instructions, and (2) the Retrieval-Augmented Generation (RAG) method will be used to enhance model performance by supplying few-shot examples similar to the input text, selected based on semantic similarity. In addition to the above-mentioned improvements, future directions of this research include (1) applying the proposed information extraction procedure to a larger set of design inspection reports, (2) standardizing the extracted information through a classification system, (3) developing a relational database to store each project's metadata alongside the extracted data, and (4) performing statistical analyses to identify patterns and trends in design defects.

References

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Info. Processing Systems, Vol. 2020-December.

Buckley, M., Zendel, A., Biggar, J., Frederiksen, L. and Wells, J. (2016). Migrant Work & Employment in the Construction Sector, ILO-UN, Geneva.

Ding, Y., Ma, J., & Luo, X. (2022). Applications of natural language processing in construction. Automation in Construction, 136:104169.

Elshaboury, H., Cecconi, F.R., Angelis, E. De, Baresi, L. and Scotti, V. (2024). NLP-based Data-Enrichment for Building Management. Proceedings of the 2024 European Conference on Computing in Construction

He, C., Yu, B., Liu, M., Guo, L., Tian, L. and Huang, J. (2024). Utilizing Large Language Models to Illustrate Constraints for Construction Planning. Buildings, 14:2511.

ISO 9000:2015 - Quality management systems — Fundamentals and vocabulary.

Jeon, K., & Lee, G. (2025). Hybrid large language model approach for prompt and sensitive defect management: A comparative analysis of hybrid, non-hybrid, and GraphRAG approaches. Advanced Engineering Informatics, 64: 103076.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, 74–81.

Marvin, G., Hellen, N., Jjingo, D. and Nakatumba-Nabende, J. (2024). Prompt Engineering in Large Language Models.

Meta AI. Llama Models, https://www.llama.com

Mistral AI. Au Large | Mistral AI | Frontier AI in your hands. https://mistral.ai/news/mistral-large.

Moon, S., Lee, G. and Chi, S. (2022). Automated system for construction specification review using natural language processing. Adv. Eng. Informatics, 51:101495.

Open AI. Hello GPT-4o. https://openai.com/index

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., et al. (2023). GPT-4 Technical Report.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J. (2002). Bleu: A Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting on Assoc. for Computational Linguistics, 311–318.

Philips-Ryder, M., Zuo, J. and Jin, X.H. (2013). Evaluating Document Quality in Construction Projects –Subcontractors' Perspective. International Journal of Construction Management, 13(3):77–94.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", Conf. on EMNLP-IJCNLP 2019, 3982–3992.

Salama, D.M. and El-Gohary, N.M. (2013). Semantic Text Classification for Supporting Automated Compliance Checking in Construction. ASCE, 30(1).

Smetana, M., Salles de Salles, L., Sukharev, I. and Khazanovich, L. (2024). Highway Construction Safety Analysis Using Large Language Models. Applied Sciences, 14: 1352.

Tilley, P.A. and McFallan, S.L. (2000). Design and documentation quality survey: comparison of designers' and contractors' perspectives.

Tilley, P.A., Wyatt, A. and Mohamed, S. (1997). Indicators of design and documentation deficiency- B.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., et al. (2023). LLaMA: Open and Efficient Foundation Language Models.

Tran, S.V.T., Yang, J., Hussain, R., Khan, N., Kimito, E.C., Pedro, A., Sotani, M., et al. (2024). Leveraging large language models for enhanced construction safety regulation extraction. ITcon, 29:1026-1038

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al. Attention Is All You Need. Advance in Neural Info. Processing Systems, 5999–6009

Xu, N., Ma, L., Wang, Li, Deng, Y. and Ni, G. (2021). Extracting Domain Knowledge Elements of Construction Safety Management: Rule-Based Approach Using Chinese Natural Language Processing. Journal of Management in Engineering, ASCE 37(2): 04021001

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y. (2019), BERTScore: Evaluating Text Generation with BERT. ICLR 2020.