Exploring the Power of Large Language Models: Automated Compliance Checks in Architecture Engineering and Construction Industries

Xiaoyu Liu

A Thesis Submitted to Cardiff University

For the Degree of Master of Philosophy

1st. January. 2025

Summary:

In recent years, Large Language Models (LLMs) have emerged as one of the most rapidly evolving sectors within the field of artificial intelligence. These models have increasingly penetrated various industries, becoming integral to our professional and daily lives. This study focuses on the customization of six specialized LLMs, each injected with professional knowledge pertinent to the Architecture, Engineering, and Construction (AEC) domain. Initially, we leveraged a bespoke integrated model to investigate the feasibility of implementing an Automated Compliance Checking (ACC) process within this industry. Subsequent phases involved employing a data-generating model for preparatory data tasks, followed by the deployment of three specialized models serving distinct target objectives. These models were interlinked to construct a prototype for the ACC process. Finally, a professional model dedicated to data analysis was utilized to quantitatively assess the performance of the entire ACC prototype in regulatory compliance checks. Drawing on the results of this analysis, we provide a comprehensive evaluation of the ACC prototype's effectiveness.

This thesis elucidates the profound potential of LLMs to revolutionize compliance checking within the AEC sector, highlighting the intricacies of developing, implementing, and refining a specialized ACC process. Through the customization and application of LLMs, this research not only showcases the practical viability of ACC but also advances our understanding of LLMs' adaptability and utility in

specialized domains. The findings contribute significantly to the ongoing discourse on the integration of artificial intelligence in industry-specific applications, offering valuable insights for future advancements and implementations.

Acknowledgements

In April of 2022, I embarked on my MPhil program journey with trepidation in my heart, unsure if I could adopt the mantle of a researcher. Yet, as my research began to take shape, I found myself increasingly captivated by the process, engaged in a labor of love that was both meaningful and fascinating. The ensuing two years have been a whirlwind, marked by an array of challenges, setbacks, and moments of doubt. Despite these hurdles, I persevered, culminating in the completion of this thesis—a testament to my journey and the stories woven through my research over these years. It is my earnest hope that it resonates with its readers.

I wish to express my profound gratitude to a constellation of individuals who were instrumental in the fruition of this research:

Foremost, I extend my deepest appreciation to my supervisor, Prof. Haijiang Li. Your unwavering support, invaluable guidance, and boundless patience provided me with the scaffold to navigate this journey. Your wisdom and mentorship propelled me forward, enabling me to transcend the myriad challenges that littered my path.

To my wife, Jin Huang, your steadfast support has been my anchor. Your enduring presence transformed the most monotonous days into opportunities for creativity and renewal. In moments of exhaustion, it was your inspiration that sparked my imagination, and in times of despair, it was your encouragement that reignited my resolve.

And to my family, whose unconditional support emboldened me to chase my passions with fervour and dedication—your belief in my aspirations has been a source of immeasurable strength.

This thesis is dedicated to each individual who has extended their support, kindness, and belief in me. Your collective encouragement has been the driving force behind my perseverance and the completion of this work. May this thesis stand as a reflection of our shared journey and the positive impacts we have fostered together in the pursuit of knowledge and discovery.

Abbreviations and Acronyms

1. ACA: Automatic Classifying Assistant;

2. ACC: Automated Compliance Checking;

3. AChA: Automatic Checking Assistant;

4. AEA: Automatic Evaluating Assistant;

5. AEC: Architecture, Engineering, and Construction;

6. AI: Artificial Intelligence;

7. AIM: Asset Information Model;

8. AIR: Asset Information Requirement;

9. ARA: Automatic Retrieving Assistant;

10. DGA: Data Generating Assistant;

11. DL: Deep Learning;

12. DS: Data Science;

13. EIR: Exchange Information Requirement;

14. GPT: Generative Pre-trained Transformer;

15. LLM: Large Language Model;

16. ML: Machine Learning;

17. NLP: Natural Language Processing;

18. OIR: Organization Information Requirement;

19. PE: Prompt Engineering;

20. PIM: Project Information Model;

21. PIR: Project Information Requirement;

22. SL: Supervised Learning.

Content

Chapter 1. Introduction

Automation in the Architecture, Engineering, and Construction (AEC) industry has gained momentum as stakeholders increasingly delegate labor-intensive and repetitive tasks to machines, thereby enhancing project delivery efficiency (Chen et al., 2018). Among these efforts, Automated Compliance Checking (ACC) has emerged as a critical component. In its broader definition, ACC refers to the use of computational tools to verify adherence to regulatory and building codes throughout the AEC project lifecycle (Thomas H Beach et al., 2013; Zhang & El-Gohary, 2016; Z. Zhang, N. Nisbet, et al., 2023). Traditionally, ACC has been implemented via rule-based or “hard-coded” methods, often requiring researchers to create digital representations of regulatory content—such as knowledge graphs in Industry Foundation Classes (IFC)—to capture semantic meaning (Badrah et al., 1998; Bouzidi et al., 2012; Garrett James & Fenves Steven, 1989; Garrett & Fenves, 1987; Sadeghineko et al., 2018). Although this approach can effectively automate certain tasks, it also demands substantial work to encode complex regulations, maintain rule sets, and ensure that design and construction documents align with them.

Early research on ACC, exemplified by Fenves et al. research in 1995, leveraged computer science techniques to improve design review efficiency (Fenves et al., 1995). However, the costs of developing and maintaining digital representations of relevant regulations, combined with limited acceptance of rigid, pre-programmed regulatory models among professional designers, hindered wide-scale adoption. Digitizing regulations proved to be an inherently cross-disciplinary challenge, requiring expertise in both computational methods and specialized knowledge of building codes.

Since 2020, however, the emergence of LLMs in NLP has opened new possibilities for ACC (Chen et al., 2024). While these AI-driven systems are not designed specifically for the AEC industry, their ability to process all types of digitized data makes them ideal for a broad range of applications. Although LLMs are computationally expensive to train, they are relatively easy to adopt, enabling researchers and practitioners to rapidly integrate them into diverse workflows(Iversen, 2024). In the context of ACC, LLMs can autonomously interpret regulatory and project documentation, reducing the need to painstakingly encode every rule or term. Major technology companies and research institutions have made state-of-the-art LLMs available at minimal or no cost, substantially lowering the financial and technical barriers for individual researchers and smaller organizations.

By harnessing LLMs, modern ACC methods can “learn” semantic meanings from text-based codes and standards, potentially simplifying the process of creating and updating digital representations. Moreover, because LLMs continue to evolve and expand their data-processing capabilities, it becomes feasible to envision future scenarios where these models serve as virtual consultants, addressing diverse and complex requirements in real time. This development marks a significant shift in how the AEC industry can approach compliance, offering adaptable, scalable solutions that can keep pace with evolving regulations and project needs.

As a researcher in the AEC industry, my goal is to leverage cutting-edge AI technologies aligned with the sector’s current digital transformation to develop a low-cost, stable, and reliable ACC methodology. In recent years, the massive amount of data generated for compliance checks has rendered manual analysis increasingly unsustainable. Modern project processes demand large-scale automated data analysis tools that can quickly interpret regulations and support informed decision-making. To achieve this, an intelligent language processing engine must either possess specialized industry knowledge or rapidly learn from relevant standards and project documentation. Such an engine requires robust semantic understanding to transfer its expertise across multiple scenarios, significantly reducing the reliance on human-driven strategies that have traditionally governed information management in many AEC projects.

There is a substantial volume of information that must be processed by various project stakeholders, which can lead to mistakes, delays, and increased risk when workloads become overwhelming. Within BIM technologies, the model itself not only contains foundational data, such as project location and dimensions, but also encompasses extensive semantic information, including building materials, construction methods, and requirements (Kumar, 2012). These details are often scattered across diverse project documents and information-model lists, creating significant challenges in terms of organization and analysis (Hassani, 2024).

In this research, original regulatory documents are fed directly into LLMs by employing PE, thereby allowing the models to autonomously interpret and learn essential knowledge from the provided documentation. Researchers then assess whether the LLMs have accurately internalized the relevant semantic content and effectively carried out compliance checks, rather than manually defining each piece of semantic information to construct a digital representation. This approach significantly enhances the efficiency of ACC by enabling a broad logical framework to be established without the need to encode every individual piece of semantic logic. Moreover, for a given set of prompts and identical files, model performance can be further improved simply by deploying more advanced LLMs, eliminating the need to retrain the entire system and thereby simplifying continual enhancements to the ACC process.

This study aims to develop a novel method in which LLMs perform batch ACC on project documents aligned with the BS EN ISO 19650-1 building information management standard. By customizing LLMs, automated compliance checks can be conducted on text data that has been extracted and refined from project documents.

The research will evaluate this entire workflow, covering the process, the LLMs themselves, and output quality at each stage, to establish a sustainable, iterative methodology capable of continuous refinement in response to evolving requirements.

Nevertheless, enabling LLMs to interpret domain-specific data, including technical specifications, regulatory standards, and industry norms, remains a significant challenge. Consequently, state-of-the-art LLMs must demonstrate the following core competencies in contextual analysis:

Domain-Specific Data Interpretation: The ability to understand and interpret specialized information, such as regulatory standards, detailed technical specifications, and unique industry practices.

Semantic Understanding of AEC Terminology: The capacity to handle the particular jargon, terms, and linguistic styles inherent to the AEC sector.

By addressing these requirements, LLMs can serve as a powerful engine for automated compliance, ultimately reducing manual effort and improving overall project outcomes.

For engineers with limited programming backgrounds, we propose a coding-free approach that leverages PE to harness the capabilities of state-of-the-art GPT models. This method enables practitioners to write scripts and implement target tasks with minimal manual coding, thereby lowering technical barriers and allowing more stakeholders to benefit from advanced AI tools.

Since 2020, LLMs have emerged as a highly popular topic across numerous fields. There is general consensus that AI-based technologies significantly improve efficiency by reducing repetitive, labor-intensive tasks (Hao et al., 2024; Pu et al., 2024). Moreover, cutting-edge AI solutions broaden the scope of problems that can be tackled, enabling users to address complex challenges via smartphones or laptops across diverse domains (Kasneci et al., 2023; Zhao et al., 2023).

Throughout the development of AI engines, data has been recognized as a critical resource, often likened to electricity in its importance for modern innovation (Naveed et al., 2023; Xu et al., 2022). One of the most prominent sub-areas of AI research is pretrained LLMs, with ChatGPT standing out as a widely known example. These pretrained models have demonstrated strong capabilities for customization and extension, effectively reducing both the cost and complexity of training or adapting LLMs for specialized applications (Teubner et al., 2023).

The AEC industry is widely considered to hold substantial potential for improvement through AI, provided that relevant data can be collected and managed responsibly. By harnessing state-of-the-art LLMs, researchers can use PE to handle highly integrated and complex tasks with relative ease. In practical terms, one could propose a scenario to GPT-4 and request a detailed, step-by-step solution for a particular problem. The GPT-4 model would then generate an implementation plan, including Python scripts and data requirements, which could be executed using the model’s API key and the supplied datasets. This approach dramatically lowers the technical barrier for individuals lacking advanced NLP expertise. Indeed, the recognition, conversion, and storage of domain-specific knowledge have never been simpler.

Nonetheless, data privacy remains a paramount concern in many AEC contexts, as stakeholders desire sufficient training data for robust AI performance while also safeguarding sensitive information. One viable strategy involves closed training—deploying models locally or within a secure, access-restricted cloud environment. This setup not only ensures the AI’s functionality but also prevents sensitive data from being inadvertently disclosed. Moreover, contemporary LLM technologies can be readily customized to address specific stakeholder requirements, and local or private-cloud storage options guarantee that data remains under strict control.

Although some scholars argue that state-of-the-art LLMs, often described as Statistics-of-Occurrence Models (SOMs) based on transformer architectures—lack true semantic comprehension or “intelligence”, these models nevertheless exhibit cutting-edge performance and remarkable potential for extension (Titus, 2024). Their core functionality lies in predicting information based on provided data, a capability that can be leveraged effectively in numerous automated data-processing scenarios.

In this research, we seek to automate the classification, labeling, retrieval, and verification of information contained in project documents, transforming unstructured text into more manageable forms such as lists, short messages, or interactive elements. More specifically, we are developing a prototype LLM-based framework for ACC in the AEC industry. This framework harnesses Deep Learning techniques to interpret domain requirements and standards, subsequently verifying project details against the relevant implementing documents. By automatically processing general industry standards, categorized into “terms,” “requirements,” and other logical groupings, the model converts these standards into easily parsable formats (e.g., JSON or Excel) before comparing the project documents against these formatted criteria. The system then provides calibration results and feedback for each compliance check. Through this approach, we aim to streamline the ACC process, reducing manual effort while maintaining a high level of accuracy.

In this research, our primary focus is the textual dimension of ACC. We demonstrate how state-of-the-art LLMs such as GPT-4 can effectively parse and verify compliance for text-based documents within the AEC context. Although this does not yet encompass the entirety of ACC, especially geometry- or model-based checks, our approach shows that, for textual information, the LLM-driven process can be taken from start to finish. As LLMs continue to evolve, they hold promise for broader integration into the complete ACC workflow, potentially covering all relevant project data in future applications.

1.1 Research route and objectives

1.1.1 Research Route

The research route of this study is depicted in Figure 1 and comprises the following key components:

1. Introduction: This chapter revolves around the application of LLMs in the AEC industry, outlining the research background, objectives, and contributions. With the rapid development of transformer-based LLMs over the past six years, these advanced AI technologies have increasingly been integrated into the data analysis domain of the AEC industry. Given the state-of-the-art LLMs' ability to assimilate professional knowledge beyond traditional NLP capabilities, they are particularly suited to automating engineering problems set against a backdrop of natural language in specific application scenarios. This study constructs an LLM-based ACC prototype capable of performing automated bulk standard checks.

2. Literature Review: In Chapter 2, our primary task is to provide a theoretical foundation for the entire study through an extensive literature review. Our examination of the literature is structured across various levels:

  a. Technical Applications of LLMs: We begin by discussing the application of LLMs from a technological standpoint, focusing on their use within the context of ACC. This discussion encompasses Compliance Check, Knowledge Interpretation, and Prompt Engineering, detailing how LLMs facilitate these aspects in the ACC process.

b. Understanding LLMs: The review then delves into "What is LLM?" Starting from the basic components, we explore Transformers and the Attention Mechanism as the foundational technologies behind LLMs. Building on this, we examine related research on LLMs, highlighting their evolution, capabilities, and the breadth of their application across various domains.

c. Extension on pre-trained LLMs: Finally, we extend our review to cover pre-trained LLMs adopting the transformer architecture that utilizes attention mechanisms. This section encompasses a broader examination of DL, NLP, ML, and AI, providing a comprehensive backdrop against which LLMs operate. Additionally, we speculate on the future of LLMs, including multimodal models, contemplating their potential evolution and application areas.

This organized layered literature review not only underpins our study with a robust theoretical framework but also maps out the trajectory of LLM development and application. It highlights the transformative potential of LLMs in the AEC industry and beyond, setting the stage for a deeper exploration of their capabilities and future possibilities.

3. Methodology: We adopt six GPT-4-based specialized models in this chapter. Unlike generic GPT-4 models found in standard ChatGPT, these models are infused with professional background knowledge and specific settings, making them more

suitable for particular tasks. The methodology is segmented into preparation, design, and implementation. The preparation phase introduces the ACC Development Assistant and DGA models for scenario exploration and simulated project file data generation, respectively. The design phase involves customizing three professional models for constructing the ACC prototype and another for its evaluation—ACA, ARA, AChA, and AEA, each serving distinct functions from classification to automatic checking. The implementation phase drives the ACC prototype with prompts, directing each model to execute its sub-task based on provided datasets, culminating in the processed results being saved and outputted as datasets.

4. Evaluation: This chapter leverages the AEA model, customized in the methodology section, to assess the results generated in the implementation phase. The evaluation encompasses not just the final outputs from the AChA model but also each process within the implementation phase, including the performances of the ACA, ARA, and AChA models. To ensure effective evaluation, the AEA model is used exclusively for quantitatively and visually assessing the other models based on clear computational rules. The aggregated quantitative assessments provided by the AEA model form the basis for a comprehensive evaluation of these models.

5. In Chapter 5, "Conclusion," we synthesize the outcomes of this study from three perspectives: "ACC Prototype Construction," "Process Implementation," and "Performance Evaluation." In the section on "ACC Prototype Construction," we detail

the characteristics and applications of six GPT-4-based specialized models: ACC Development Assistant, DGA, ACA, ARA, AChA, and AEA. These models were integral to developing a robust ACC prototype, each serving a unique function within the broader framework of ACC in the AEC industry.

In "Process Implementation," we discuss the strategies employed to drive LLMs using PE, aimed at circumventing potential pitfalls inherent in LLMs. This discussion is categorized into three key areas: "Avoiding Vague Outputs and Regurgitation," "Reducing Hallucinations," and "Optimizing Prompt Outputs." Each category represents a set of tactics designed to enhance the clarity, reliability, and efficacy of LLM outputs, ensuring the models' responses are precise, coherent, and directly applicable to the tasks at hand.

6. Chapter 6, building upon the insights from Chapters 4 and 5, encapsulates the primary methodologies adopted in constructing, driving, and evaluating the ACC prototype. It lays out a future research trajectory aimed at more closely integrating the downstream applications of LLMs into addressing engineering problems within the AEC industry. This forward-looking stance emphasizes the desire to deepen the involvement of LLMs in the sector, leveraging their advanced capabilities to tackle complex challenges and streamline processes.

Through this structured conclusion, the thesis not only highlights the significant achievements of the research but also sets the stage for future exploration and application of LLMs in the AEC industry. It underscores the potential for LLMs to revolutionize industry practices, offering a blueprint for ongoing innovation and implementation in the realm of ACC and beyond.

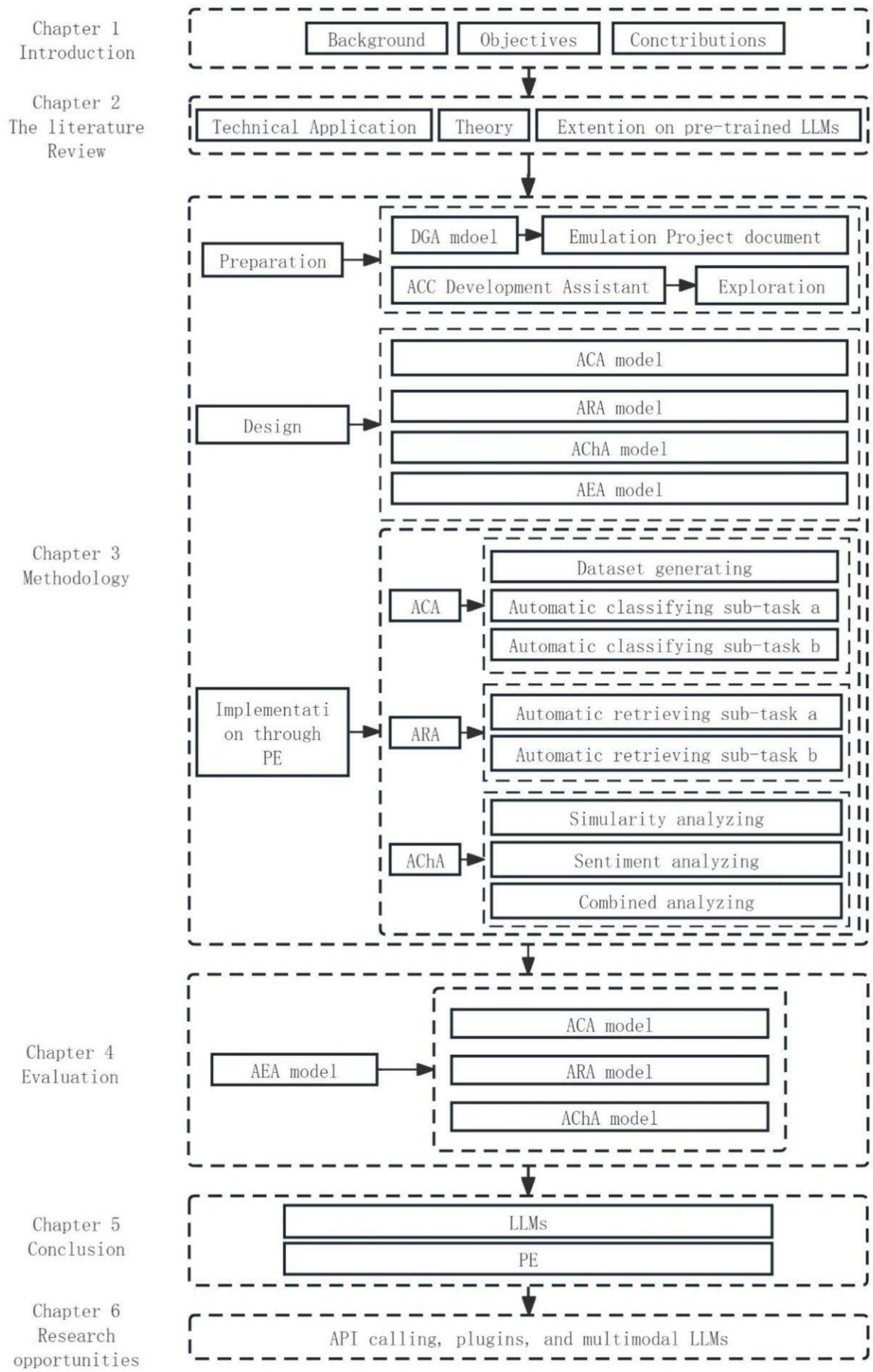

Figure 1. Research route of ACC in the AEC industry.

1.1.2 Objectives

In this study, our research objectives encompass the following aspects:

1. Exploration of LLMs' Application Potential in the AEC Industry: We delved into the capabilities of LLMs within the AEC sector, ranging from utilizing these advanced models for research to addressing specific NLP tasks inherent to the industry.

2. Comprehensive Implementation Strategy for ACC using LLMs in the AEC Industry: We investigated the full implementation plan for ACC within the AEC industry. This included the overarching design philosophy and the intricate details of specific tasks.

3. Customization of Specialized GPT-4 Models to Construct an ACC Prototype: We tailored professional GPT-4 models to develop an ACC prototype aimed at enabling a codeless, ACC process within the AEC industry.

4. Development of Specialized GPT-4 Evaluation Models: These models were designed to assess the performance of the ACC prototype in executing ACC sub-tasks, offering a quantitative measure of its effectiveness.

Through these objectives, our research not only aims to unveil the potential of LLMs in revolutionizing the AEC industry's approach to compliance checking but also to provide a blueprint for the practical application and evaluation of such models in addressing complex industry-specific challenges.

Chapter 2. The Literature Review

According to the development progress of LLMs, all of the relevant technologies within 4 years starting from 2020 are introduced and reviewed from specific engineering technologies within the scenarios of ACC to general architectures and principles of machine learning algorithms. There are several parts which are specifically reviewed as essential elements of the process.

Firstly, we are going to review the advent and development of ACC in the AEC industry and combine them with the state-of-the-art AI based technologies which are deployed to implement this specific scenario through LLMs named PE. This technology allows researchers to implement automatic batch, repetitive processing to the target datasets with less expertise of coding and data processing. The PE can drives LLMs which have billions of parameters with very limited scale of datasets that only contain essential domain knowledge and information of relevant scenario. In additional, the comprehensive review of the LLMs is executed in terms of architectures, deployment, development, and operation etc. The language models like GPT which adopt transformer architecture and billions of parameters have lots of names to describe the features from various perspectives. Though the applications' scenarios are quite similar, they generate sentences according to the provided monologue. In today's research about these models, fewer scholars doubt the performance of the models but most of them start developing the potential possibilities of downstream applications based on LLMs to address tasks in complex semantic environments. Finally, we conclude the current LLMs relevant technologies which can be deployed on the ACC scenario and the future research opportunities of LLMs which can achieve the scenario more sophisticated and comprehensive.

2.1 Technologies of ACC in the AEC Industry

2.1.1 Compliance Checking

"The compliance checking process occurs constantly throughout all phases of a project lifecycle in the AECO (Architecture, Engineering, Construction, and Operation) domain and affects all aspects of the lifecycle" (Amor & Dimyadi, 2021). It is usually processed by the expert of the relevant industry as there are numerous of domain knowledge required. Normally, there are 2 types of ACC processing underpinning data sources. The first type is based on the text message from the project including the project contracts, brief, specifications, construction analogs, progress and handing over documentation etc. The second type is graphical including blue prints and building information models etc. (Amor & Dimyadi, 2021; Joao et al., 2021; Sagar et al., 2015).

- **Automated Compliance Checking**

The common way of executing the ACC process is explicit transferring the standards through hard coding then align the project documentation with the codes’ requiring forms (Joao et al., 2021; Malsane et al., 2015). An another method is labelling the labelling the standards’ clause and the building models, then match the clauses and corresponding elements with the same label and execute the checking process (Thomas H. Beach et al., 2013; Cesarotti et al., 2014; Greenwood et al., 2010). In the realm of ACC, various methods have been developed that offer a degree of automation. However, a notable limitation of these approaches is their reliance on specific document formats, which restricts their applicability (Ilal & Günaydın, 2017; Zhang & El-Gohary, 2016). Additionally, these methods are not dynamically adaptable to changes in standards; they require manual intervention in the form of hard coding to update the rules in accordance with any revisions to the standard (Soliman-Junior et al., 2021). Furthermore, traditional methods of ACC are primarily effective with standards that can be formalized or quantified computable. These methods fall short when dealing with criteria that defy such formalization (Ilal & Günaydın, 2017). Furthermore, they are highly sensitive to the format of the content that requires formulaic assessment, often leading to errors and discrepancies (Salama & El-Gohary, 2011). Lastly, conventional ACC methods are not capable of directly analyzing text documents. This necessitates manual preprocessing of the documents, a task that is both time-consuming and labor-intensive. Moreover, such manual intervention cannot guarantee accuracy.

The automated systematic documents information retrieval methodologies can be learned from the research of Kruiper et al., combined with Query Expansion and Document Expansion, an information retrieval system has been developed, which addressed the tasks of information retrieving and matching during ACC process . The research of Kruiper et al. provide a practical way of implementing aligning semantic meanings of documents query with the corresponding regulations (Kruiper et al., 2023; Kruiper et al., 2024). Through the ifc, the methodologies can be applied comprehensively to any other documents applying the same information classification.

- **Traditional methods of ACC**

"The quest to automate compliance checking processes needed for planning, design, construction and operations has been an active research topic for half a century" (Amor & Dimyadi, 2021). Prior to the widespread application of Transformer-based pre-trained models like BERT and GPT, as noted in research by Amaral et al., the implementation of ACC relied on conventional NLP methodologies. This involved transforming text into machine-readable data through word embedding techniques, followed by semantic tagging. The process then used various machine learning algorithms (such as the classical MLP algorithm) for recognition and categorization based on these tags. Finally, these steps were translated into methods that enabled machines to understand semantics. This traditional approach to ACC was considerably more dependent on

explicit semantic tagging and machine learning classification algorithms prior to the advent of advanced, context-aware models like BERT and GPT. (Amaral et al., 2021).In research of Zhou et al. in 2022, the methods of checking text documents are based on Context-Free Grammar (CFG) which requires appropriate preparing before executing checking processing. Generally, the CFG methodology contains 4 steps, preprocessing, semantic labelling, syntactic parsing and rule generation (Z. Zhang, N. Nisbet, et al., 2023; Zhou et al., 2022). In our review of ACC, we understand that the primary method for this application involves converting target regulations and documents to be checked into a logic that can be understood by machines. The process then entails syntactic analysis to annotate the logic with various tags for classification. Following this, the tagged rules are used to query and match the documents under review. Finally, the ACC is executed, and the results are outputted. The most challenging task in this process is the syntactic analysis and tagging. Training language models to deeply understand the semantics of text is a complex and resource-intensive endeavor. It requires a substantial amount of high-quality data, intricate algorithms, and considerable computational power. These requirements often exceed the resources available to most research teams. Consequently, traditional methods for exploring ACC have been very limited, typically deployed in very specific application scenarios. Moreover, without regular updates in line with changes in regulations, the effectiveness of these methods diminishes over time. Therefore, the high costs, limited research, and restricted application scenarios constitute the current state of automated checking. We posit that the current approach of aligning document formats for automated checking remains a time-consuming and labor-intensive process. It does not, in a true sense, enhance efficiency but rather shifts the time and labor costs from the checking process to the preliminary stages of document preprocessing, content parsing, and format alignment. Our aspiration is to develop a more comprehensive form of ACC, one that does not rely on prior manual annotations and processing, and can be executed by machines without such interventions.

- **AI-based methods of ACC**

Since 2022, the rapid development of deep neural network algorithms based on the Transformer architecture has propelled large language models, such as the GPT series, into the spotlight. Numerous applications based on these models have been developed and deployed across various aspects of the networked virtual world(Chang et al., 2023). However, it is regrettable that research on ACC systems based on large language models remains limited, and the market lacks such systems. On the other hand, this area, as a downstream application field of large language models, represents an unexplored 'blue ocean'. More promisingly, the rapid advancement of large language models has dramatically reduced the costs associated with research and development in their downstream applications.

The need for costly hardware investment and extensive manually annotated datasets has been significantly diminished. Now, a single individual with a portable laptop and a dataset annotated by a handful of experts can drive a large language model with over a billion parameters. Such setups can achieve impressive performance in customized application scenarios. Since 2022, the rapid development of deep neural network algorithms based on the Transformer architecture has propelled large language models, such as the GPT series, into the spotlight. Numerous applications based on these models have been developed and deployed across various aspects of the networked virtual world. However, it is regrettable that research on ACC systems based on large language models remains limited, and the market lacks such systems. On the other hand, this area, as a downstream application field of large language models, represents an unexplored 'blue ocean'. More promisingly, the rapid advancement of large language models has dramatically reduced the costs associated with research and development in their downstream applications. The need for costly hardware investment and extensive manually annotated datasets has been significantly diminished. Now, a single individual with a portable laptop and a dataset annotated by a handful of experts can drive a large language model with over a billion parameters. Such setups can achieve impressive performance in customized application scenarios.

2.1.2 Knowledge Interpretation

The advancement of large language models, particularly in the realm of ACC, has revolutionized the process of knowledge integration, a critical subtask in this domain. Historically, knowledge integration required manual, labor-intensive methods, such as constructing semantic webs, extracting regulatory clauses, and transforming regulatory logic into machine-executable code (Guo et al., 2021; Martinelli et al., 2019; Zheng et al., 2022). Techniques such as semantic tagging, truncation, and conversion were essential in enabling machines to recognize and process specialized knowledge (Cejas et al., 2023; Salama & El-Gohary, 2013; Zhang & El-Gohary, 2021). However, the emergence of sophisticated large language models, like GPT-4, has significantly streamlined this process, reducing the manual effort and cost associated with it (Nguyen et al., 2023). These models inherently understand natural language semantics, facilitating easier and more efficient integration of domain-specific knowledge into ACC systems. The current approach involves utilizing prompt engineering and model fine-tuning, a significant evolution from the prior manual methodologies. This evolution allows for more effective batch ACC across multiple project documents, enhancing the efficiency and accuracy of the compliance checking system (Beach et al., 2020; Ratnayake & Wang, 2024; Z. Zhang, L. Ma, et al., 2023). Consequently, the adoption of these advanced large language models in the knowledge integration process signifies a pivotal shift towards more automated, efficient, and accurate compliance checking in various domains (Sharma & Yegneswaran, 2023).

- The Development of the Methods of Knowledge Interpretation

In the realm of downstream applications of large language models, two primary methods of knowledge integration stand out: model fine-tuning and prompt engineering (Voetman et al., 2023). Fine-tuning is a process of customizing large language models by adjusting their parameters using specially designed datasets. This tailors the model to perform better in specific application scenarios. For instance, fine-tuning GPT-3 with domain-specific data enables the generation of more accurate and relevant outputs within that domain. The fine-tuning process involves altering the model's internal weights to reflect the characteristics of specific types of data and queries. This method requires a substantial amount of annotated data and computational resources but can significantly enhance the model's performance on specific tasks (Yiheng Liu et al., 2023).

On the other hand, prompt engineering leverages the existing knowledge and capabilities of a model, guiding it to produce desired outputs through carefully crafted prompts. With the enhanced processing abilities of advanced models like GPT-3.5 and GPT-4, the number of tokens they can handle has increased considerably, allowing for prompts that include more details and specialized knowledge. Prompt engineering does not modify the model itself but relies on effectively utilizing the model's pre-trained capacities. Well-designed prompts can steer the model to understand and respond to complex queries and handle professional or domain-specific issues, even without specific fine-tuning. This method is more resource and data-efficient, offering greater flexibility for large-scale applications.

In summary, fine-tuning and prompt engineering each have their advantages: the former adapts the model to specific tasks, while the latter leverages the model's general capabilities for diverse queries. As large language models continue to advance in processing capabilities, the potential for these methods in applications like ACC is steadily growing.

2.1.3 Prompt Engineering

In the field of AI, a 'prompt' typically refers to the input for a large language model, encompassing all the information needed to drive the model to perform a target task. These inputs are generally composed of natural language, which is the language used by humans in everyday scenarios. Initially, prompts were primarily in English, but now, the latest large language models can accept prompts in multiple languages. Moreover, these models have evolved to accept multimodal input, including voice, images, and even dynamic videos. Unlike traditional programming languages, prompts do not require strict formatting of input statements. However, prompts that are simple in structure and logical in their composition can enhance the quality of the model's outputs. The design and generation of effective prompts often require experimentation, summarization, and iterative refinement (Marvin et al., 2023).

The concept of prompts in LLMs highlights the shift from rigid programming syntax to more flexible and natural forms of human-machine interaction. As language models evolve, they become increasingly capable of understanding and responding to a diverse array of inputs, moving beyond text to include other modes of communication, thereby broadening the scope and applicability of AI technologies.

In general, each DL model needs to be trained by large-scale datasets to regress massive parameters before the advent of the large-scale pre-trained DL models with billions of parameters . Over time, the topic has undergone tremendous evolution, evidenced by a number of noteworthy advancements and research papers. PE has become an essential study for harnessing the potential of LLMs (Fan et al., 2023; Yi Liu et al., 2023; Muktadir, 2023). It's noteworthy that the methods of constructing prompts in prompt engineering also evolve with the development of large language models. Early models, such as GPT-1, GPT-2, and the unrefined version of GPT-3, lacked full semantic understanding capabilities. They primarily mimicked based on the content of the prompts, leading to an early focus on Few-Shot Learning in prompt engineering, where examples are added to prompts to help the model understand and execute the target task. However, with the advancement of large language models, as seen in the latest models like GPT-3.5, GPT-4, and LLaMa, these models have gained the ability to understand more complex semantic information and can execute target tasks directly from prompts. Consequently, recent prompt engineering primarily revolves around Zero-Shot Learning. Furthermore, the increased capacity of current models supports the inclusion of extensive professional knowledge within prompts, enhancing their ability to perform tasks in specific application scenarios.

In summary, while model fine-tuning and prompt engineering remain the primary methods of integrating knowledge into large language models, the approach within prompt engineering has shifted from Few-Shot to Zero-Shot Learning, reflecting the models' enhanced understanding capabilities. This evolution has opened up new possibilities for applying these models in fields like ACC, where the nuanced understanding of complex professional knowledge is crucial.

PE is an emerging study from 2022 that strives to enhance the performance of LLMs' output (White et al., 2023). Initially, PE wasn't a formalised study but a part of skills which can drive the pre-trained models together with the skill of fine-tuning as the early-stage LLMs were rough and didn't generate complex comprehension to a monologue, the models act more like imitation than understanding the execution.

Additionally, the use of prompts in the latest large language models varies across different environments. Taking GPT-4 as an example, there are primarily three environments in which it operates: the default environment (web interface), the Copilot environment, and the programming environment (API). Each of these environments has its unique characteristics. In the default environment, GPT-4 has the widest applicability and the least stringent requirements for prompts. Here, the language model can directly process and generate pure text files (such as documents or data tables), including simple analysis and processing of provided data (such as annotation), or generating images without complex semantics.

The Copilot environment functions akin to a search engine embedded with artificial intelligence. It can utilize prompts to query content using Microsoft's Bing search engine, enabling browsing, filtering, and summarizing of content, and ultimately providing users with more precise information. The programming environment, on the other hand, supports embedding GPT-4 as an API into programs for automated data analysis and batch processing.

However, large language models are not without their imperfections, and certain inherent risks can lead to biases in the execution of target tasks.

Due to the inherent lack of interpretability in the structure of deep neural networks, it is challenging to accurately judge the output of large language models. Studies have shown that these models may produce 'hallucinations,' which are outputs that seem plausible at first glance but do not withstand scrutiny. As a result, large language models often underperform in generating creative content. To ensure stable and reliable output, explicit constraints through prompt engineering are necessary. This includes designing rigorous deductive logic, providing reliable references, and setting checkpoints for manual review at critical junctures.

LLMs are pre-trained models, developed using vast amounts of unlabelled data that is mostly sourced from the web without any cleaning process. This unfiltered data can contain disturbing, unsafe, and biased content, which the models inevitably learn. Despite considerable efforts to limit the generation of unsafe, biased, or disturbing content by these models, a small number of users have reported encountering such issues during their use, development, or research of large language models. Therefore, when designing prompts, it is crucial to implement appropriate safety clauses or utilize publicly available, proven safe datasets to minimize such occurrences.

In summary, these different environments for GPT-4 demonstrate the model's versatility. From handling a wide range of tasks in the default environment to providing AI-enhanced search capabilities in the Copilot environment, and facilitating automated processing in the programming environment, GPT-4's applications are diverse and adapt to various user needs and technical requirements (Shin et al., 2023). While large language models represent a significant advancement in AI, their limitations and potential risks cannot be overlooked. Issues related to interpretability, content reliability, and inherent biases in training data necessitate careful handling and constraints in their application.

- **Few-Shot Learning of the Prompt Engineering**

This feature encourages emerge of “Few-Shot Learning,” which is a simpler type of Prompt Engineering applied to LLMs to drive the model. Few-Shot Learning refers to the Prompt that provides several samples to let the models learn the patterns of text generating in specific scenarios. For example, reading comprehension on early-stage models is a typical Few Shot Learning: the prompt includes a random chunk of monologue and several pairs of questions and corresponding answers, finally following with a single answer and letting the models generate the answer. This type of prompt is largely applied on BERT, GPT-1, GPT-2, and GPT-3 etc. (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, et al., 2020).

- **Zero-Shot Learning of the Prompt Engineering**

The other type of prompt is “Zero-Shot Learning,” (also known as “In-context Learning”) which means that the input only contains instructions, explanations, and descriptions etc. but without any samples provided, the language models can execute the tasks entirely based on the comprehension of the prompt. This type of prompt requires the language models to possess the semantic understanding of the unstructured data, this task is usually implemented by the fine-tuned language models based on artificially labelled datasets (Devlin et al., 2018; Radford et al., 2018; Radford et al., 2019; Reynolds & McDonell, 2021).

Through the power of the state-of-the-art LLMs, most of tasks can be implemented by PE. There are magnitude industries that apply PE to their terrains for executing text information parsing tasks, i.e. information assistants, smart programmers, automated labelling etc. With the development of the LLMs, it is easier and taking less time for researchers to learn to drive LLMs, and as the LLMs present great capabilities of text generating on most of the text datasets, researchers can apply PE and small scales of datasets to LLMs to their own scenarios. This means that the researchers focus more on their domain knowledge and integrate their domain knowledge into the LLMs through PE and domain knowledge fine-tuning datasets. Though most of LLMs at early stage cannot understand or appropriately execute the prompt of Zero-Shot Learning before fine-tuning and further processing like Reinforcement Learning from Human Feedback (RLHF). Through novel LLMs like GPT-4, the PE can implement direct unstructured data batch processing by generating codes and internal prompts to achieve the whole ACC process (Fan et al., 2023; Mosbach et al., 2023).

Considering the complexity of the LLMs, it is challenging to explain the work within the neural networks explicitly, it is hard to find the absolute best prompt to drive the LLMs in a specific scenario but always emerges better prompts which could be the relatively optimal prompt within a certain range.

- **LLMs Evaluation**

In the evaluation of Large Language Models (LLMs), methodologies are broadly categorized into two main types: automatic evaluation methods and human evaluation methods.

Automatic evaluation methods are chiefly employed to analyze quantifiable metrics such as accuracy, recall, and F1 scores, which are standard parameters for gauging performance (Chen et al., 1998; Liang et al., 2022). In addition, the confusion matrix is a commonly used tool for the visual representation of automatic evaluations. Beyond these, metrics like calibrations, fairness, and robustness are also significant indicators. According to definitions in Chang et al.'s research,

"Calibrations pertain to the degree of agreement between the confidence level of the model output and the actual prediction accuracy.

Fairness refers to whether the model treats different groups consistently, i.e., whether the model's performance is equal across different groups.

Robustness evaluates the performance of a model in the face of various challenging inputs, including adversarial attacks, changes in data distribution, noise, etc.

" (Chang et al., 2024)

Given that these metrics often follow uniform rules and can be precisely calculated through equations, they offer a greater level of objectivity. This allows for batch processing across numerous LLMs, datasets, and tasks, yielding objective and stable evaluation outcomes. However, in specific scenarios, automatic evaluation methods may not fully appraise the performance of LLMs, potentially overlooking crucial metrics that cannot be obtained through calculation. Therefore, to comprehensively and sophisticatedly evaluate LLMs' performance in scenarios, it is imperative to incorporate results from human evaluation methods.

Human evaluation methods are employed to assess tasks involving LLMs that cannot be precisely calculated, such as complete summarization tasks, output generation based on prompts, and complex reasoning tasks (Zhao et al., 2024). These tasks require the application of relevant professional background knowledge to evaluate the LLMs' performance in various scenarios. Human evaluation methods strive to minimize subjectivity in the assessment process by aligning evaluation criteria, similar to setting unified evaluation standards in related research, including accuracy, relevance, fluency, and safety.

Accuracy measures the semantic alignment between the LLMs' outputs and the reference outputs, akin to precision in logic (Kadavath et al., 2022).

Relevance assesses the semantic relatedness of the LLMs' outputs to the reference outputs, analogous to recall in logic .

Fluency gauges the coherence of continuous output from LLMs in response to prompts, focusing on the continuity of semantic context.

Safety evaluates the degree of risk associated with the LLMs generating harmful content (inappropriate, discriminatory, harmful, etc.) based on prompts.

Employing human evaluation methods to assess LLMs' performance in target tasks within scenarios offers greater flexibility, a more specific evaluation framework, and clearer results. However, these evaluations must be grounded in rigorous and aligned criteria to maintain objectivity (Singhal et al., 2023). In this context, sometimes leveraging results from automatic evaluation methods as a basis for assessment can be a wise choice.

2.2 The Theory of LLMs

2.2.1 Transformer and Attention Mechanism

Since the introduction of the attention mechanism in 2015, the field of NLP based on deep learning has adopted a new architecture beyond CNNs and RNNs, known as the Transformer architecture. The Transformer consists of two components: encoders and decoders. Both of these components can replace RNNs in recording the positional encoding of sequential input data, making them integral to the core of deep neural networks. Under this architecture, two major language models were initially developed: Google's encoder-based BERT and OpenAI's decoder-based GPT.

Transformers, a groundbreaking deep learning model architecture for processing sequential data, were developed in 2017 by Vaswani et al. from the Google team. This innovative work, based on the attention mechanism, has significantly influenced subsequent trends in machine learning research. Its importance and impact have even surpassed seminal works involving CNNs and RNNs. Since 2018, numerous Transformer-based algorithms, including BERT, GPT, LlaMa, GLM, Bard, Gemini, and others, have been developed and applied across various industries.

Apart from the encoders and decoders, another distinguishing feature of networks based on the Transformer architecture is their size and depth. Ranging from millions to billions of parameters, researchers have continually expanded the size of language models. It is from this period that language models based on the Transformer architecture and equipped with millions of parameters or more have been referred to as 'large language models.'

The advent of the attention mechanism and the Transformer architecture marked a significant shift in the landscape of deep learning for NLP. This era saw the emergence of more complex and capable models, such as BERT and GPT, which have significantly influenced the field with their unprecedented scale and depth.

The first introduction of the transformer architecture was in 2017 by Vaswani et al., this work improved the architecture of RNN but made it simpler, and improved the efficiency while executing NLP tasks (Vaswani et al., 2017a). In today research, the DL algorithms with the best performance are all based on the transformer's architecture.

In 2022, Transformer-based large language models were still in their 'youthful' stage and not as 'intelligent' as the models we see today. At that time, the pioneering models mainly included the encoder-based BERT and the decoder-based GPT-1 and GPT-2. Comparing the performance on various public datasets, BERT generally outperformed GPT-1 and GPT-2, swiftly becoming a popular research direction in the NLP field. Meanwhile, the OpenAI team continued to expand and train the GPT series, culminating in the billion-parameter GPT-3 model. OpenAI innovatively applied Reinforcement Learning from Human Feedback (RLHF) techniques to GPT-3, leading to the development of GPT-3.5. This approach significantly enhanced the model's capabilities. As a result, terms like GPT, Transformer, and NLP became some of the hottest buzzwords in the AI field during 2022-2023.

In this study, we extensively applied and deployed large language models within the context of ACC, covering aspects such as process design, task segmentation, dataset construction, and data analysis. However, due to constraints related to costs (including time, financial, and labor), this study did not involve building and training Transformer-based deep learning networks from scratch. Instead, we employed publicly available and safe pre-trained models. The data and datasets used for research and demonstration purposes were also derived from publicly available sources, converted into simulated data and datasets. The primary reason for conducting a literature review on the Transformer architecture is to accurately analyze and explain the principles of large language models, elucidating their training, operational, and deployment methods. This approach allows for a more comprehensive and cautious assessment of the effectiveness of using large language models in this specific application scenario.

Before delving into the specifics of the Transformer architecture, it is important to briefly explain the concept of the attention mechanism. The purpose of the attention mechanism is to allocate weights to each part of the input data, indicating the importance of each part in the current context. These attention weights are dynamically calculated based on the input, meaning that the model focuses on different parts for different inputs. Within the Transformer architecture, the predominant form used is the multi-head attention mechanism, first introduced by Vaswani et al. in their 2017 paper on the Transformer architecture. Unlike traditional single-head attention, where the model can only learn the relationships in input data from one perspective, multi-head attention allows the model to understand data from multiple perspectives simultaneously. This is achieved by running multiple independent attention mechanisms in parallel. The advantage of this approach is that each 'head' can focus on different features and relationships within the data. The parallel structure of multi-head attention also enables the model to learn various types of information concurrently, thus enhancing efficiency and performance.

“Unlike earlier self-attention models that still rely on RNNs for input representations, the transformer model is solely based on attention mechanisms without any convolutional or recurrent layer. Though originally proposed for sequence-to-sequence learning on text data, transformers have been pervasive in a wide range of modern deep learning applications, such as in areas to do with language, vision, speech, and reinforcement learning” (A. Zhang et al., 2023). Transformers rely entirely on attention mechanisms, particularly Multi-Head Self-Attention, to extract features from sequences, eliminating the need for recurrent layers. A Transformer

typically comprises two main parts — the encoder and the decoder, each consisting of multiple identical layers stacked together. The encoder processes the input data, while the decoder generates the output. Without recurrent structures, Transformers use positional encoding to maintain the position information of words in a sequence. Unlike RNNs that process data sequentially, Transformers can process entire sequences in parallel, significantly improving computational efficiency. RNNs often struggle with capturing long-distance dependencies (i.e., the relationships between inputs and outputs that are far apart). Transformers effectively address this issue with self-attention mechanisms. Due to parallelization and the avoidance of complex recursive computations, Transformers generally train faster than RNNs.

Advantages of Transformers:

High Efficiency: Their ability to process data in parallel makes Transformers more efficient in handling large datasets.

Scalability: They are well-suited for large-scale training tasks.

Flexibility: They can be adapted for a wide range of applications.

Driving Methods of Transformers: Pre-training and Fine-tuning.

Pre-training:

Dataset: Pre-training typically requires a large-scale, diverse corpus. For example, BERT uses Wikipedia and BooksCorpus, while GPT utilizes an even larger corpus of web text.

Cost: Pre-training is costly, requiring significant computational resources and time. The larger the model, the higher the cost.

Pre-training Tasks: Common pre-training tasks include Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

Fine-tuning:

Workload: The workload for fine-tuning is relatively small since the model's structure and parameters have already been initialized through pre-training.

Dataset Preparation: Specific small datasets tailored for particular tasks (such as sentiment analysis, question answering, etc.) are needed.

Main Approach: During fine-tuning, the model's final output layer is often modified to suit a specific task, and it is trained on a dataset for that particular task. The learning rate is usually set lower to preserve the knowledge acquired during pre-training.

Figure 2. The Transformer – model architecture illustrates the original development of the Transformer architecture by Vaswani et al. This architecture consists of an encoder and a decoder, each handling the input and output aspects, respectively. We will now explain in detail each part of this architecture. In general, the Transformer learns the content of the input using a multi-head self-attention

mechanism and predicts the subsequent content as output. Considering its primary application in the field of NLP, the input in this study is pure text composed of natural language.

Encoder: The encoder consists of N (in the original paper, N=6) identical layers, each with two sub-layers. The first step is preprocessing the input text, where the text is converted through word embedding, followed by positional encoding based on the input sequence. Then, it proceeds to the two formal processing sub-layers. The first sub-layer is a multi-head self-attention processing layer, where the preprocessed content is passed through a multi-head self-attention layer with residual connections, followed by normalization (Ba et al., 2016), constituting the first sub-layer's processing. The second sub-layer is a fully connected feed-forward layer, taking the output from the first sub-layer's multi-head self-attention, passing it through a fully connected layer with residual connections, and then another round of normalization, resulting in the output of the second sub-layer. Repeating this two-sub-layer processing six times completes the encoder's output.

Decoder: As shown in Figure 1. The Transformer – model architecture, the decoder consists of N (in the original paper, N=6) identical layers, each comprising three sub-layers. Unlike the encoder, the decoder receives inputs from two directions: the output from the previous moment and the content transformed by the encoder. The decoder incorporates an additional masked multi-head self-attention sub-layer, where the output from the previous round, after preprocessing (word embedding plus positional encoding), enters through residual connections. This sub-layer, unlike the multi-head attention sub-layer in the encoder, features a critical operation of adding a mask to the output of the previous round. This ensures that the decoder does not 'see' words that have yet to be generated in the self-attention mechanism, thus maintaining the autoregressive nature of the decoder. Notably, at the start of outputting the first word, since there is no output from the previous round, the output after the residual connection is zero. The input processed by the encoder directly enters the decoder's multi-head self-attention sub-layer, while the output from the previous round's masked multi-head attention sub-layer enters the decoder's multi-head self-attention sub-layer for processing and normalization through residual connections. The output from this multi-head self-attention sub-layer then passes through a feed-forward neural network sub-layer within the decoder for further processing and normalization. After repeating the processing of these three sub-layers six times, the output is fed into a fully connected layer and then into a softmax layer to produce the probability distribution of the sequence output (Vaswani et al., 2017a).

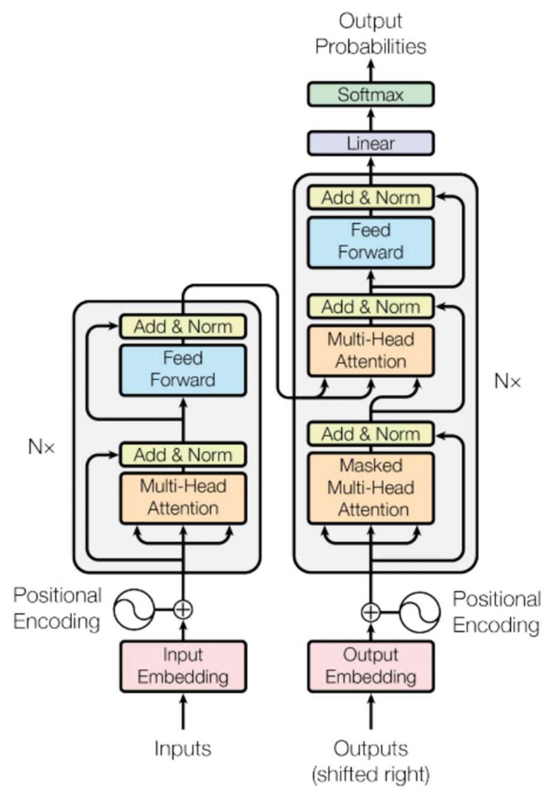

Figure 2. The Transformer – model architecture (Vaswani et al., 2017b)

The Transformer model architecture later became the cornerstone for large language models and even multimodal models. Subsequent models (apart from BERT, which adopted the decoder's architecture) primarily utilized only the encoder part of the architecture, discarding the decoder. On this foundation, these models underwent extensive pre-training with massive datasets. Then, based on the pre-trained models, they were fine-tuned with more complex and advanced tasks using manually annotated data. This approach led to the development of some of the most advanced large-scale models in existence today, such as GPT-4, Gemini, and others.

2.2.2 Large Language Models

"Language Models (LMs) are computational models models that have the capability to understand and generate human language. LMs have the transformative ability to predict the likelihood of word sequence or generate new text based on a given input. Large Language Models (LLMs) are advanced language models with massive parameter sizes and exceptional learning capabilities" (Chang et al., 2023). Researchers were trying larger Neural Networks on transformers, from millions parameters to hundreds billions of parameters. With such magnitude networks, more and more data, even knowledges can be stored into the models, in that case, researchers introduced the idea of "Pre-trained Model," and "Large Language Models (LLMs)." Basically LLMs are included 2 types basic stages, pretrained models and finetuned models which are based on different types of training data and applying layers. Most of LLMs provide open source pre-trained models for people to execute further study for customizing, and few academic or commercial institutions publish finetuned models for applying, the most representative models are BERT, GPT, Llama etc. These models have the similar capabilities of processing text information. There are many others of terms to describe the LLMs like GPT from various of perspectives to explain this type of models' features including architectures, task orientation, training methodologies, information types, generations or versions, applications etc. (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, & Askell, 2020; Cho et al., 2014; Devlin et

al., 2018; Ng & Jordan, 2001; Radford & Narasimhan, 2018; Ramesh et al., 2021; Vaswani et al., 2017a).

2.2.3 Deep Learning

“The world needs deeper and wider Neural Networks.” This is an imaginary slogan that I made up. But it was a trending that in that time researchers were developing larger NNs and improving the efficiency as they could. The first Deep Learning Neural Network algorithm was developed by Yann et al. which was called LeNet, an improved Convolutional Neural Network (CNN) based on AlexNet by Krizhesky et al. which was largely applied in Computer Vision area (Krizhevsky et al., 2012; Lecun et al., 1998). The type of DP algorithm which actually processed sequential data was Recurrent Neural Network (RNN) as known as Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997). In hence this architecture significantly improved the performance on NLP. But still, the models can implement very limited functions during this stage.2.6 Pretrained models

Pretrained Models usually refer to the LLMs which have been trained by the large scales of unlabelled data. A LLM needs to be trained by large scales of data before applied. In idea case, the provided data is clean and has high quality so the training process can be easy and fast. But the cost of improving data quality is too much, so researchers split the training process into 2 sub-tasks.

Firstly, let the LLMs are trained by large scales of “low quality” data. This data are collected from everywhere, and just roughly jumbled together and to be feed in to the LLMs. This process is called “pre-training.” And we get a “Pre-trained model” after this initial process. Pre-trained models will have the parameters initially set, in case study of GPT-3 series model, the essential function of the pre-trained model is imitating according to the provided prompts, it is also known as “few shots learning,” which means a model can fast learn patterns from the prompts. However it is hard to say the model has the knowledge at this stage as the model can’t comprehend or answer most of the instructions of the prompts whatever it is a single request or chain-of-thoughts. The datasets for pre-training are exceptionally large and diverse (about 500 billion tokens in total) so they can generally learn the initial parameters of large language models (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, & Askell, 2020). The models which complete the pre-training process are ready for fine tuning and further customizing.

2.2.4 Natural Language Processing

“*Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications*” (Liddy, 2001). NLP has a long history in AI technology, emerging as an interdisciplinary field of AI and linguistics since the 1950s. Initially, NLP and information retrieval were distinct disciplines, with the former used for text translation and the latter for indexing and searching vast text repositories. However, with technological advancements, these fields have gradually merged. AI-based NLP primarily follows two approaches: rule-based and algorithm-based. The main tasks in NLP can be broadly categorized as follows:

Text Retrieval: Retrieving relevant information from target texts based on provided keywords and instructions.

Document Classification: Categorizing provided text content, such as sentiment analysis, which is a subcategory of document classification.

Auto-Completion: Completing the remainder of a text based on provided content, which could be the rest of a sentence or an entire article.

Language Translation: Translating provided text content into other languages.

Document Annotation: Performing syntactic analysis and annotating target texts based on provided instructions.

Text Summarization: Summarizing and abstracting provided text content.

Text Inference: Understanding the semantics of provided texts to generate specific answers with semantic information for target questions.

Given the complex background and the need to integrate multiple task modules in NLP applications, it is often challenging to categorize real-world application scenarios under a single NLP task. In other words, NLP applications typically require more detailed subdivision and redefinition according to specific instance requirements and provided data, rather than being hastily categorized into a single simple task. Taking this study as an example, the application scenario discussed is ACC. In this scenario, we need to correspond the checking regulations with the relevant clauses of the documents under review, set checking instructions using prompts, and then execute the check through a large language model to output the results as 'compliant' or 'non-compliant.' This process is a composite task, which we divide into four simpler sub-tasks: labeling, classifying, retrieving, and checking. By integrating these simple tasks through prompts, we achieve ACC.

This research adopts a novel methodology of NLP based on Prompt Engineering of LLM. In the earlier study of NLP, text documents and datasets need to be converted before training which is called word embedding. Word embedding is a technology that convert unstructured data like text into structured data which can be read by AI models. After processed by word embedding, the datasets can be feed into AI language models. The most representative word embedding technologies are Word2Vec, GloVe, FastText, ELMo, and transformer-based Models like BERT and GPT, etc. (Church, 2017; Kenton & Toutanova, 2019; Pennington et al., 2014; Radford et al., 2018; Shahbaz et al., 2019; Wu & Manber, 1992).

The typical NLP contains three sections of processes: Firstly the documents are converted into datasets which have structured data. Then the datasets are feed into the language process engine to execute target tasks and get the results datasets in forms of structured data. Finally the results datasets are converted back to the unstructured data which can be read by people. This procedure is still adopted by state-of-the-art NLP technologies including GPT, LlaMa, Bard etc.

2.2.5 Machine Learning

"A machine learning algorithm is a computational process that uses input data to achieve a desired task without being literally programmed (i.e., "hard coded") to produce a particular outcome. These algorithms are in a sense "soft coded" in that they automatically alter or adapt their architecture through repetition (i.e., experience) so that they become better and better at achieving the desired task" (El Naqa & Murphy, 2015). Basically it means an algorithm that the mathematical model can learn itself the weights and bias from comparing the difference value between prediction and actual results, and it can reduce the difference through iteration. Based on this idea, a large number of ML algorithms were developed in past decades. During this term, there are magnificent number of ML algorithms emerged. Furthermore, *"the adoption of data-intensive machine-learning methods can be found throughout science, technology and commerce, leading to more evidence-based decision-making across many walks of life, including health care, manufacturing, education, financial modeling, policing, and marketing"* (Jordan & Mitchell, 2015). (Samuel, 2000)

And during that time, Neural Network was developed by David E. Rumelhart et al (Rumelhart et al., 1986). This is an extraordinary work but didn't bring too many contribution at that time as that was a crazy time too many discoveries were found at that time. But still there were many researchers followed his steps as they scoped the potential of this development.

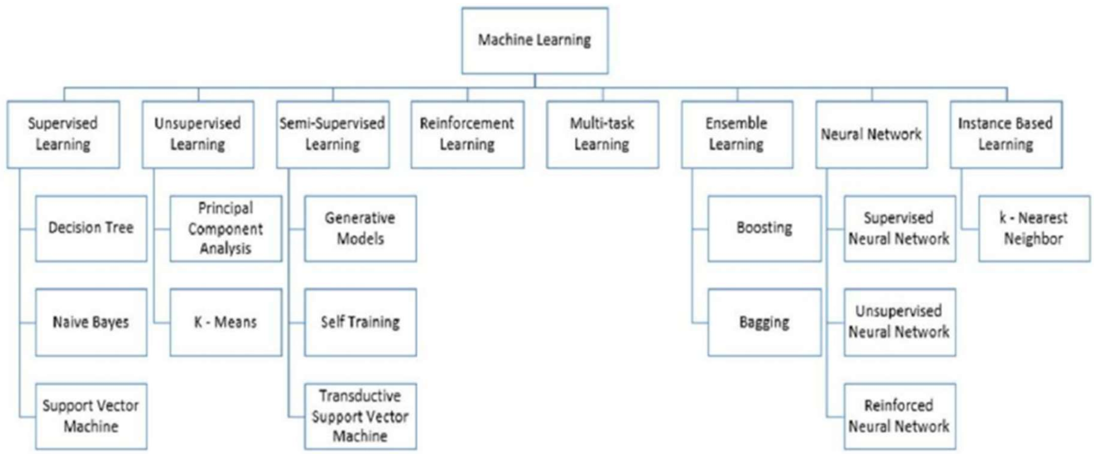

Figure 3. Basic classification of general ML algorithms (Mahesh, 2020).

The term 'Machine Learning' first appeared in Samuel's research in 1959 and subsequently saw significant development in 1965. From Mahesh's review in 2020 (Figure 3 is cited from this document), we can observe the 'Basic classification of general ML algorithms.' According to this review, up until now, machine learning algorithms have been divided into eight categories: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning, Multi-task Learning, Ensemble Learning, Artificial Neural Networks, and Instance-Based Learning.

Supervised Learning: *”Supervised learning entails learning a mapping between a set of input variables X and an output Y and applying this mapping to predict the output for unseen data”* (Cunningham et al., 2008).

Unsupervised Learning: Unsupervised Learning is a type of Machine Learning algorithms trained by unlabeled data, which means the algorithms are required to recognize patterns without any guidance (Barlow, 1989; Ghahramani, 2003).

Semi-Supervised Learning: “*Semi-supervised learning is the branch of machine learning concerned with using labelled as well as unlabelled data to perform certain learning tasks. Conceptually situated between supervised and unsupervised learning, it permits harnessing the large amounts of unlabelled data available in many use cases in combination with typically smaller sets of labelled data*” (Van Engelen & Hoos, 2020).

Reinforcement Learning: *“Situated in between supervised learning and unsupervised learning, the paradigm of reinforcement learning deals with learning in sequential decision making problems in which there is limited feedback”* (Wiering & Van Otterlo, 2012).

Multi-task Learning: *“Multi-Task Learning (MTL) is a learning paradigm in machine learning and its aim is to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks”* (Zhang & Yang, 2022).

Ensemble Learning: *"Ensemble learning methods exploit multiple machine learning algorithms to produce weak predictive results based on features extracted through a diversity of projections on data, and fuse results with various voting mechanisms to achieve better performance than that obtained from any constituent algorithm alone"* (Dong et al., 2020).

Artificial Neural Network: "Artificial Neural Network (ANN) is a hot topic in artificial intelligence since the 1980s. It abstracts the human brain neural network from the perspective of information processing, establishes a simple model and compose different networks according to different connections" (Wu & Feng, 2018).

Instance-Based Learning: *"Instance-based learning (IBL) algorithms are a subset of exemplar-based learning algorithms that use original instances from the training set as exemplars. One of the most straightforward instance-based learning algorithms is the nearest neighbour algorithm. During generalization, instance-based learning algorithms use a distance function to determine how close a new input vector* $\vec{y}$ *is to each stored instance, and use the nearest instance or instances to predict the output class of* $\vec{y}$ *(i.e., to classify* $\vec{y}$*)" (Wilson & Martinez, 2000)*.

2.2.6 Artificial Intelligence

Distinct from machine learning, AI is a broader concept, coined by John McCarthy in 1956 (McCarthy et al., 2006). Up to now, AI applications have penetrated nearly every field, including intelligent search, recommendation systems, intelligent audiovisual processing, generative AI, and gaming. The development of AI has experienced its ups and downs until the emergence of deep learning in 2012, which drew widespread attention due to its remarkable performance. The introduction of the Transformer architecture in 2017 marked a significant milestone, leading to a global AI boom that swept across various industries by 2020.

2.3 Extension on pre-trained LLMs

From the initial large language models such as GPT-1 (Radford et al., 2018), BERT (Devlin et al., 2018), and LLaMA-1 (Touvron, Lavril, et al., 2023) with hundreds millions of parameters, to later models like GPT-3.5 (Ouyang et al., 2022) and LLaMA-2 (Touvron, Martin, et al., 2023), and the most recent advancements in GPT-4 (Achiam et al., 2023) and Gemini, these models have not only made qualitative leaps in text processing capabilities but also significantly broadened the scope of data sources and types they can handle. This expansion enhances researchers' efficiency in exploring, developing, and deploying LLM applications, making a more substantial contribution across various industries. This section reviews the development trends of several widely recognized LLMs, based on their expanded data source access and data processing capabilities, and briefly outlines the expected changes in these trends over the next year.

2.3.1 Scope of Data Sources

Early pre-trained LLMs lacked internet access, limiting their output to non-real-time information. Due to constraints on the range of training datasets, these models lacked specific industry knowledge, making it challenging to answer complex technical questions. The essence of this problem is LLMs' inability to autonomously acquire specific datasets based on prompts. Researchers have adopted two different paradigms to address this issue:

a. Developer Paradigm: This approach embeds internet and search engines into LLMs, allowing them to actively search for relevant data based on prompts. Typical cases of this paradigm include applications developed on GPT-4, such as the current version of ChatGPT and Copilot. This paradigm is suitable for obtaining data from public sources through fuzzy searches, often resulting in large volumes of data that may lack precision.

b. User Paradigm: Users directly provide LLMs with datasets containing targeted information. Through prompts, LLMs retrieve relevant information from these datasets, resulting in outputs that feature the dataset's specific data characteristics. Typical applications of this paradigm include Retrieval Augmented Generation, with libraries like LangChain and Lamini providing specific retrieving datasets for LLMs like LLaMA and GPT (Lewis et al., 2020). ChatGPT also supports this functionality, allowing users to upload datasets to inject professional knowledge into ChatGPT. This paradigm produces high-quality content when dealing with prompts involving complex professional knowledge. However, the scales of datasets provided by users are generally smaller than those obtained through the developer paradigm, limiting the model's generalization capabilities when processing similar prompts.

2.3.2 Types of Data Processed

The transformer architecture was initially developed to better handle sequential data, specifically to address NLP problems (Vaswani et al., 2017a). Following the success of NLP using transformer-based LLMs, researchers began exploring the use of these models in the Computer Vision field for various tasks, including image classification, object detection, semantic segmentation, instance segmentation, and image generation (Chen et al., 2021; Dosovitskiy et al., 2020; Prabhakar et al., 2024; Song et al., 2021; Strudel et al., 2021). Of particular interest are the research outcomes combining computer vision and NLP fields into "multimodal models." These transformer-based pre-trained models can process both text and image information within the same large model. Multimodal models can recognize various semantics in images, such as objects and implied information, and generate text or images as output based on prompts. Notable examples of multimodal models include GPT-4, PaLM, Perceiver IO, DALL·E, CLIP, and LaMDA. Essentially, 2024 is set to witness significant transformations in LLMs' functionalities and applications, focusing on creating AI systems capable of understanding and generating diverse human-like content. These advancements will redefine interactions between humans and machines, opening new possibilities for AI applications across various fields.

Chapter 3. Methodology

3.1 Overview

This section comprehensively elaborates on how this research has studied the development of large language models (LLMs) over the past two years, showcasing the improvements in model performance within this application scenario as a result of the evolution of LLMs themselves. Through the use of Prompt Engineering (PE) and Fine-Tuning (FT), automation in standard checks for fire protection design in architecture and project information management in the Architecture, Engineering, and Construction (AEC) industry has been achieved for related text documents.

In the early stages of our research, we explored various NLP pathways including Long Short-Term Memory (LSTM) models, encompassing early hardcoding techniques, syntactic parsing, and the classical deep learning algorithm LSTM. However, these techniques were not well-suited for our research for two main reasons: firstly, the experiments required for our study could not obtain data from open-source datasets, necessitating the generation of datasets manually. Due to limited resources, it was not possible to obtain datasets with a large number of samples, and datasets with a small number of samples could not sufficiently train an

LSTM model; secondly, our research aimed to adopt a general approach, but methods such as hardcoding and syntactic parsing lacked strong generalization capabilities. Therefore, after reviewing machine learning (ML) methods, we began experimenting with the most advanced deep learning (DL) models for the task of automated standard checks, specifically those based on the transformer architecture, like the large predictive models.

After selecting LLMs as our primary research pathway, our study progressed in two parts: initially, we evaluated the semantic understanding capabilities of early LLMs (GPT-3 and GPT-3.5), confirming that these models' abilities were sufficient for the most crucial sub-task of automated standard checks, "checking". Subsequently, with the release of OpenAI's latest GPT-4 model, its semantic understanding capabilities were significantly enhanced. After a detailed comparison of the capabilities of the GPT-4 model, we constructed a complete and simpler prototype for automated standard checks based on GPT-4. By decomposing the complex task into several simpler tasks, using only prompt engineering, and relying on the outputs of each task with GPT-4 to retrieve and verify results, we achieved a complete automation of standard checks.

Throughout our research process, we went through roughly four stages: Initially, upon our first engagement with large language models, the lack of understanding regarding these models led us into a prolonged period of exploration. This phase of exploration continued as the development of large language models rapidly progressed. After gaining some understanding of the capabilities of large language models, we commenced planning for the application scenario of automated standard checks. This involved limiting the processing content of the large language models to pure text, decomposing complex tasks into a combination of simpler subtasks, and finally integrating the data from each subtask to perform the checking tasks.

Subsequently, based on our plan, we prepared the corresponding data for experiments and collected the results. Lastly, we evaluated the experimental outcomes. If the results were unstable or did not meet the requirements, we would return to the planning stage to adjust the process and prompts, conduct experiments, and collect data again, until the results were acceptable or tended to stabilize. At this point, the process would be terminated, and we would confirm the performance of the large language model in the current application scenario at this stage.

3.2 Preparing

Before formally beginning the construction of the entire ACC processing prototype, we undertook two preparatory tasks:

The first preparatory task involved using prompts, project files, and related standards to establish a GPT-4 custom exploration model for the development of the ACC prototype. This customized GPT-4 model was primarily aimed at exploring the comprehensive capabilities of GPT-4. We injected a significant amount of professional knowledge related to our study into the model to enhance its ability to accurately understand professional terminology within conversations. The outputs from this customized model would serve as the main reference. The model was tasked with addressing a range of requirements in the research, including but not limited to exploring the boundaries of the model's semantic understanding capabilities. For instance, we directly tasked the model with implementing complex comprehensive requirements and recorded the feedback from the GPT-4 large language model. Additionally, the model was used to test simple tasks that involved professional knowledge, such as summarizing monologues, classifying terms within project documents, querying datasets based on keywords, matching keywords, and ultimately performing checking functions through semantic understanding.

The second task involved customizing a GPT-4-based professional model for data generation. In early research phases, substantial resources were expended on manually generating data. Consequently, in later research stages, we shifted towards employing customized GPT-4 professional models to automatically generate simulated project file data based on prompts. This strategic pivot to automated data generation using GPT-4 significantly streamlines the process of creating large, diverse datasets necessary for training and testing the models. By automating this step, we not only conserve valuable resources but also enhance the scalability and adaptability of the research, allowing for a broader exploration of scenarios and more robust model validation. Customizing GPT-4 for specific data generation tasks leverages the model's advanced capabilities to produce realistic, complex datasets that closely mimic real-world data, providing a solid foundation for subsequent stages of the ACC process.

In Chapter Three, "Methodology," we introduce three methods for customizing models through the injection of professional knowledge. The first method is fine-tuning, which we applied to the early GPT-3 series models. We fine-tuned the GPT-3 pretrained model using .json files converted from artificially generated simulation datasets and successfully implemented the checking subtask in ACC with the fine-tuned model. The advantage of this method is significant when working with early LLMs with limited capacity, as it can substantially reduce the number of prompt tokens required. Additionally, fine-tuning significantly enhances the early LLMs' semantic understanding capabilities for target tasks. Fine-tuning typically modifies the parameters of LLMs, usually affecting the last layer, which means the data we provide has a profound impact on the model. In other words, the large language model will, to some extent, "remember" the input background knowledge. The drawback of this approach is the need to construct a fine-tuning dataset, which usually requires considerable manual effort to ensure high quality.

The second method involves directly using prompts to inject professional background knowledge into LLMs with sufficient capacity. This approach can be applied for quickly handling specific simple tasks. Its limitation is that it usually targets a particular processing step without generalizability. Furthermore, directly using prompts to inject professional background knowledge does not alter the model's parameters, meaning the model will not "remember" the provided professional background knowledge in different dialogues. Each new dialogue requires re-entering the relevant professional background knowledge in the prompts.

With the evolution of LLMs, we have gained a third method for injecting professional background knowledge into LLMs. In the latest GPT-4 series models, the process of knowledge injection has been simplified and can now be directly conducted through OpenAI's ChatGPT interface using prompts and various file formats (such as .pdf, .txt, .xlsx, .csv, .json, etc.) to customize the GPT-4 model. This method allows for rapid and effective customization of LLMs for a range of roles, output formats, and application scenarios while learning from the provided files. This learning process can change the parameters within the large language model to some extent, achieving an effect similar to fine-tuning. Thus, the model can "remember" the professional background knowledge to a certain degree, but the operation is much simpler.

3.2.1 GPT-4 models customization

In this study, we primarily utilized OpenAI's GPT-4 Large Language Models (LLMs). By iterating through prompts, we explored and gradually developed executable plans based on the output results from GPT-4. In the initial phase, we customized a model for automated standard checks within the AEC (Architecture, Engineering, and Construction) industry using GPT-4 as the foundational model. The advanced capabilities of the GPT-4 model allowed us to add specialized knowledge and presets through prompts and text files (such as .csv, .pdf), setting the context of text content understanding and reasoning within the backdrop of automated standard reviews in the AEC industry, thereby defining the scope of standards to be examined. As demonstrated in the subsequent prompts, we can directly describe the role that the LLM needs to play in the application scenario, including industry background, specialized knowledge, target tasks, text types to be processed, modes of thinking, and output content. This pre-sets the process reasoning and final outputs of LLMs, allowing for an initial customization of GPT-4.

Additionally, we can add professional background knowledge to GPT-4 by uploading text files, such as professional terminology, processing procedures, and requirements found in standards. This knowledge can now be directly fed into the GPT-4 model, where it is understood and learned by the large model. Finally, we uploaded the collected files required for model learning directly into the LLMs, allowing the GPT-4 model to automatically learn. Researchers then observed GPT-4's learning performance in subsequent tests and further updated the GPT-4 model by providing more data files through prompts or creating new GPT-4 task models, integrating them in programs or dialogue windows.

Prompt for customizing a GPT-4 model for general practices of ACC process implementation.

User prompt: You are a helpful assistant of ACC in the Architecture Engineering and Construction industry. In this project, you are required to help people develop a comprehensive automated compliance-checking process for project documents according to the provided standards. There are several sub-tasks you will execute in this project based on the development of the workflow and corresponding pipelines, including workflow developing, pipeline building, code interpreting, document converting, prompt reviewing, knowledge ingesting etc.

Following the initial customization prompts for GPT-4 LLMs, the GPT builder engages in multiple rounds of dialogue with researchers to ensure the model is correctly customized and can appropriately execute the target task. The first step in this process involves the GPT builder requesting us to name this customized model.

GPT builder: How about we call this GPT Compliance Architect? Does this name work for you?

User prompt: I prefer the name of ACC development assistant.

Following the naming of the customized GPT-4 model, the GPT builder would proceed to generate an icon for the customized GPT-4 model and seek feedback from researchers. Researchers can provide suggestions for modifications to the icon through prompts. For instance, in this experiment, the model initially generated an icon as shown in Figure 4, labeled as the original icon.

Figure 4. Original icon of the customized GPT-4 LLM.

GPT builder: Here's the initial profile picture based on your GPT's theme. Do you like it, or would you like any changes?

User prompt: It looks good, can you slightly change the figure behind the tick? I want it to be a figure of documents, papers or books.

Afterward, the model generated a new icon and once again sought our feedback. Following several rounds of modifications, we finalized the icon as shown in Figure 5, labeled as the final version icon.

Figure 5. The final version icon.

GPT builder: I've updated the profile picture to a flat design without shadows, featuring a stylized AI processor chip and a checkmark. This design focuses on a clean and contemporary look, ensuring it's premium and easily identifiable at a small size. How does this final version look to you?

User prompt: Yeah, that is the final.

GPT builder: Great! Now that we've finalized the profile picture, let's move on to refining the context for the ACC Development Assistant.

What specific types of documents or standards should this assistant focus on for compliance checking?

Subsequently, GPT inquired about the details of the standards referenced. Given that GPT-4 now supports the uploading of files in formats such as .csv and .pdf, and the model can recognize and understand the graphical information within these files, in this study, we uploaded both the transformed dataset files of the standards to ensure GPT-4's fundamental understanding of the standards, and the original standards documents in .pdf format. This was an attempt to provide GPT-4 with professional background knowledge related to the graphical information, such as various pipelines and flowcharts.

User prompt: A project information management standard in the AEC industry named ISO 19650, I uploaded the .csv dataset to present the content of this standard.

Finally, the GPT-4 model automatically processes and updates the summary of the learned content within the "instruction" section, integrating the acquired knowledge and adjustments based on the provided files and feedback. This ensures the model is accurately tailored to the specific application requirements, incorporating both the basic understanding of the standards and the professional background knowledge from graphical information.

“You are a helpful assistant specializing in ACC within the Architecture, Engineering, and Construction (AEC) industry, focusing on ISO 19650, a standard for project information management. Your role is to assist users in developing comprehensive automated compliance-checking processes for project documents, ensuring adherence to ISO 19650 standards. Your tasks include workflow development, pipeline building, code interpretation, document conversion, prompt reviewing, and knowledge ingestion. You offer guidance, technical support, and insights on creating

and optimizing workflows and pipelines for compliance checking, ensuring project documents align with ISO 19650 standards and regulations. You have access to detailed information from ISO 19650-1 and ISO 19650-2 documents, including graphical details, to provide more informed assistance.”

After customizing the large language model, we can initially input simple prompts along with prepared simulation project files and the standard dataset from public data sources into the customized GPT-4 model for a trial run. Although it's unlikely that the model will directly fulfill such complex requirements, the customized GPT-4 will analyze the demands and attempt to process the provided dataset. Upon analyzing and determining that the current prompts and dataset information are insufficient for direct implementation, the GPT-4 model will offer additional suggestions, including acquiring more information and more detailed processing, to seek solutions.

In the initial stage, we designated the following prompt as our preliminary prompt, within which we articulated our task requirements in the most succinct manner possible. Subsequently, we received the ensuing feedback from GPT-4 (the preliminary testing was conducted through the assistant in the playground).

User prompt: help me implement the requirement of ACC to provided project documents according to the provided standard documents.

Run instructions: *You are a helpful assistant specializing in ACC within the*

Architecture, Engineering, and Construction (AEC) industry, focusing on ISO 19650, a standard for project information management. Your role is to assist users in developing comprehensive automated compliance-checking processes for project documents, ensuring adherence to ISO 19650 standards. Your tasks include workflow development, pipeline building, code interpretation, document conversion, prompt reviewing, and knowledge ingestion. You offer guidance, technical support, and insights on creating and optimizing workflows and pipelines for compliance checking, ensuring project documents align with ISO 19650 standards and regulations. You have access to detailed information from ISO 19650-1 and ISO 19650-2 documents, including graphical details, to provide more informed assistance.

ACC Development Assistant: *To effectively assist you in implementing the ACC of your project documents according to ISO 19650 standards, it's essential to break down the process into actionable steps. This will involve:*

Understanding ISO 19650 Standards*: First, we need to have a clear understanding of the requirements laid out in the ISO 19650 standards, particularly ISO 19650-1 and ISO 19650-2 concerning the management of information over the whole life cycle of a built asset.*

Analyzing Project Documents*: Before we can check for compliance, an analysis of the types of project documents you are working with is crucial. This involves* *identifying the content, format, and how these documents are typically structured and used within your projects.*

***Defining Compliance Criteria**: Based on the ISO 19650 standards and the types of project documents, we will define specific compliance criteria. This step involves mapping out standards requirements to aspects of your project documents.*

***Developing a Compliance Checking Workflow**: With the criteria in hand, we will outline a workflow that automates the process of checking your project documents against the compliance criteria. This workflow would involve processes for document ingestion, analysis, reporting, and potentially remedies for non-compliance.*

***Implementing the Workflow with Tools and Technologies**: We'll identify and leverage the appropriate tools and technologies to build the compliance checking pipeline. This could involve document management systems, natural language processing (NLP) tools, and custom scripts for automated checks.*

***Testing and Iteration**: After the implementation, the system will go through a series of tests with real project documents to ensure that it accurately detects compliance and non-compliance. Iterations will likely be needed to refine detectors and processing based on the test outcomes.*

***Deployment and Training**: Once the system is deemed reliable, it will be deployed. This stage may also involve training for your team members on how to interact with the compliance checking system and interpret its outputs.*

If you have specific sections of ISO 19650-1 and ISO 19650-2, or project documents you are already working with and require direct assistance on, please provide these materials or details so I can offer you more personalized guidance.

User prompt: I uploaded 2 .pdf documents, can you process them?

Detailed_skyscraper_contract_2.pdf

BS EN SIO 19650-1-2018.pdf

ACC Development Assistant: *Thank you for clarifying that the provided documents are in PDF format. With this information, I can now proceed to extract the text content from these PDF documents, which will allow us to analyze the ISO 19650-1 dataset and template project documents more thoroughly.*

Let's start by extracting the content from both PDF documents. This will enable us to understand their structure and contents better, forming a basis for developing the ACC system according to ISO 19650 standards.

ACC Development Assistant: *The content extraction from the ISO 19650 PDF document seems successful, providing an introduction that indicates it's the correct document: "BSI Standards Publication" related to "Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling. Part 1: Concepts and principles."*

However, it appears there was difficulty accurately extracting meaningful text from the second PDF document containing the template project documents, as the output contains non-standard characters and might be due to the PDF's structure or encoding.

Given this situation, we can proceed in two ways:

Focus on the ISO 19650-1 document: We can analyze the extracted content from the ISO 19650 document to define compliance parameters, suggest a feasible approach for compliance checking, and outline a structure for implementing it.

Address the Project Document Extraction Issue: We can try a different method or tool to extract the content from the second PDF, potentially enabling us to directly utilize your template documents for defining specific compliance checks based on actual document content.

Please let me know how you would like to proceed or if there's any specific aspect of the ISO 19650-1 document you want to explore first.

User prompt: I would like you to learning all of the terms and domain knoledge, and split all of the requirements from ISO 19650-1. Then you can use the domain knowledge to check the clauses of project documents reach the requirements, if they don't, please tell me which clause is not reached.

ACC Development Assistant: *To address your request thoroughly, we'll follow a multi-step approach. Firstly, we will focus on the ISO 19650-1 document, extracting and* *learning all of the terms and domain knowledge it contains. Subsequently, we will identify and list all the requirements stated within. Upon establishing a comprehensive understanding and list of ISO 19650-1 requirements, we will then analyze the project documents, comparing their clauses against the ISO 19650-1 requirements to identify any areas of non-compliance.*

During the preparation phase, in addition to manually created simulation project file datasets, we also utilized a customized GPT-4 to automate the generation of simulated project files. This customized GPT-4 model can be created through ChatGPT's interface in a codeless manner, significantly lowering the barrier for researchers without a coding background to utilize Large Language Models (LLMs). The following prompt demonstrates how we customized a professional GPT-4 model through simple prompts and dialogues.

The content below details the prompt used to construct a customized GPT-4 for generating a simulated project file dataset, along with a dialogue with the GPT-4 AI. The dialogue has been abridged, primarily omitting discussions about generating icons and naming, as these aspects have been presented in previous comprehensive models.

Prompt for customizing a GPT-4 model for generating emulation project document data.

GPT Builder: *Hi! I'll help you build a new GPT. You can say something like, "make a* *creative who helps generate visuals for new products" or "make a software engineer who helps format my code."*

What would you like to make?

User prompt: Prompt for customizing GPT-4 of generating emulation project document datasets: In this research, we are going to develop a comprehensive ACC process to project documents according to the project information management standards in the Architecture, Engineering, and Construction industry. You are a helpful assistant in generating emulation project documents according to the provided prompt and regulation documents.

The model has to generate the emulation project document against the knowledge within the regulation documents and the specifications of the prompt. In most of the time, the generated project documents' format is .doc. The model also needs to take further tasks of document transferring and analyzing.

GPT Builder: *For this GPT, how about the name DocuGen Architect? Does that work for you?*

User prompt: DGA would be ok.

GPT Builder: *Great! Now, let's refine DGA's behavior to ensure it meets your needs exactly.*

To start, let's consider its role and goal a bit more. Should DGA provide detailed guidance on how to ensure documents meet specific regulatory standards, or focus *more on generating documents based on user inputs?*

User Prompt: yes.

GPT Builder: To clarify, DGA will provide detailed guidance on ensuring documents meet specific regulatory standards and focus on generating documents based on user inputs.

Next, let's talk about what DGA should emphasize or avoid in its interactions. Are there any particular phrases, technical jargon, or details it should specifically use or steer clear of?

User prompt: It will be provided in future prompts as every time the requirement is slightly different

GPT Builder: Understood. DGA will adapt its responses based on the specific requirements provided in future prompts, ensuring versatility and accuracy in generating and analyzing documents.

How about the way DGA communicates? Would you prefer a more formal tone for professional settings, or a slightly informal approach to make interactions more engaging?

User Prompt: Formal tone would be good. But you can always ask me any details of emulation document generating during the process if you feel the provided information is not enough.

GPT Builder: I've updated DGA to communicate in a formal tone and to be open to asking for additional details during the document generation process if the provided information isn't enough.

DGA is now set up to assist you with generating and analyzing emulation project documents in the AEC industry, tailored to each unique prompt you provide.

Would you like to test DGA in action, or is there anything else you'd like to adjust?

User prompt:

[action]*BS EN ISO 19650-1-2018.pdf* document provided.

The .pdf document is the general document contains the general domain knowledge of the industry. You are required to take the document as the essential domain knowledge reference.

GPT Builder: *I've updated DGA to use the BS EN ISO 19650-1:2018 standard as its essential domain knowledge reference. This document outlines the principles for information management in the AEC industry, focusing on Building Information Modeling (BIM) to manage and produce information throughout the lifecycle of built assets.*

DGA now has a solid foundation for generating and analyzing emulation project documents, ensuring compliance with project information management standards. It's ready to assist you with document generation, analysis, and compliance checks *in a formal and professional manner, tailored to the specific requirements you provide.*

Would you like to test DGA with a specific task now, or is there anything else you'd like to adjust?

User prompt: That's all at the current stage. I might modify the model in the following test sections.

GPT Builder: *Great! DGA is all set up and ready to assist you with your emulation project document needs in the AEC industry. Remember, you can always come back to refine or modify its behavior based on your experiences during testing.*

Feel free to start testing DGA whenever you're ready, and let me know if there's anything you'd like to adjust or any new requirements you have. I'm here to help make DGA as effective and tailored to your needs as possible.

By this point, we have completed the initial customization of a proprietary GPT-4 model for generating simulated project files. Next, within this customized model, we will use prompts and corresponding standard documents to generate simulated project file data, making certain modifications to the model during the operational process. This step involves not just the generation of data but also iterative refinements to the model to ensure its outputs are closely aligned with the requirements of the task, demonstrating the adaptability and flexibility of the GPT-4 model in meeting specialized application needs.

We have named our GPT-4 model the Document Generating Architecture (DGA) model. With the initial customization settings for the DGA model complete, we now aim to generate the required simulated project files from public data sources using prompts to drive and test the large language model.

Before presenting the conversation with the DGA model, let's first briefly analyze the purpose of this session and the information that needs to be conveyed to the DGA model. This prompt is primarily composed of three parts: the target task, the explanation of the standards dataset, and an introduction to the simulation project.

Target Task: Typically, at the start of the conversation, we directly set the target task for the GPT-4 model in the prompt and specify the role the model is to perform. In this research, we have set the target task as, "Generate project information management contracts compliant with the simulated project files' standards requirements."

Standards Dataset Explanation: Although we injected knowledge into the model by directly uploading the original files of ISO 19650 during the model's customization, to effectively reduce hallucinations, we processed the original .pdf files of ISO 19650-1. This process included removing images from the original document and extracting purely textual terms. We then added two different types of features to the dataset and explained the classification method and specific categories, ultimately producing a .csv format dataset.

Emulation Project Descriptions: *From public data sources, we extracted basic information from several projects. After removing sensitive information, we created descriptions for the simulation projects and used these descriptions to guide the DGA model in generating simulated project file data. This approach ensures that the model has a clear understanding of the task at hand, informed by a concrete dataset and specific project contexts, thereby facilitating the creation of realistic and relevant project documents.*

After the Document Generating Architecture (DGA) model has been customized, we will delve into how we utilized the DGA model along with task prompts to generate simulated project data in the subsequent section 3.1.3. These data will be employed for future testing purposes. With this, our work in the preparation phase concludes. Next, we will describe research related to infusing knowledge into Large Language Models (LLMs), encompassing fine-tuning and further customizing interconnected LLMs according to specific tasks. This phase is crucial for enhancing the models' understanding and processing capabilities, ensuring they are well-equipped to handle the complexities of ACC processes with a high degree of accuracy and efficiency.

3.1.2 Fine-Tuning

To ascertain whether early stage LLMs (GPT-3, GPT-3.5) possess the requisite semantic understanding capabilities essential for performing the core functions of automated standard checks, we initially conducted a series of tests based on fine-tuning and prompt engineering to verify the models' capacity for concrete execution of standard checks. According to OpenAI's documentation, early iterations of large language models, such as GPT-3, lacked the intrinsic semantic understanding to be directly prompted. Instead, they required fine-tuning with a small sample dataset to comprehend the semantics of the fine-tuning dataset, thereby processing prompts with similar structures. The paramount task during dataset preparation involved manually generating a multitude of "checking prompt - checking result" pair samples. Theoretically, the larger the dataset, the more accurate the outcomes produced by the large language model. However, in situations of limited resources, the model can also be fine-tuned with datasets comprising approximately a few hundred samples to understand prompt content and generate acceptable outcomes. In this study, we initially extracted 104 standards solely for text checks from the HTM 05-02 specifications, creating corresponding design specification clauses for each standard, including at least one positive example that meets the standard and one negative example that does not. Each example was manually classified as either "met" or "not met," forming a simulated dataset of "checking prompt" and "checking result" pairs. This dataset was then convert into .json format files, inputted into the LLMs for fine-tuning, thereby generating a customized LLMs tailored to the application scenario of this study.

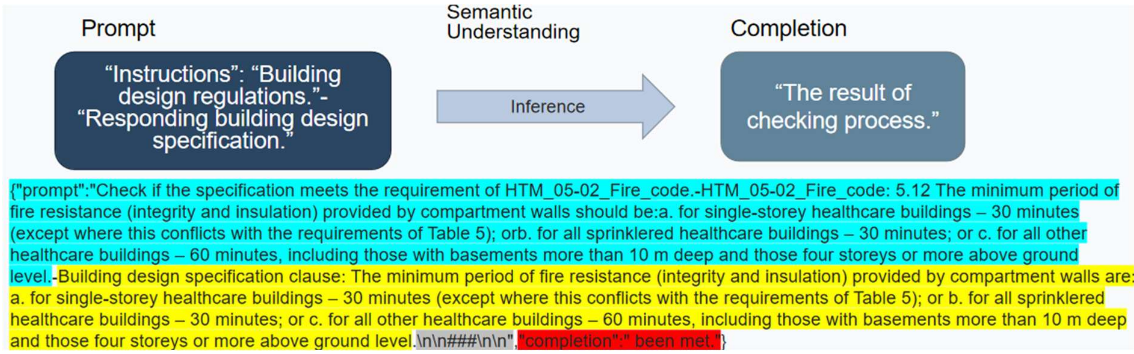

Figure 6. An example of a “checking prompt” and corresponding “checking result” pair.

Fine-tuning has been a crucial method in our early endeavors to enrich models with professional knowledge and enhance their semantic understanding capabilities. In our preliminary research, we were able to expand the functionalities of early GPT models, initially limited to rephrasing, to encompass reasoning abilities. For example, our investigations into the GPT-3 and GPT-3.5 models revealed that employing small datasets consisting of several hundred samples could significantly improve GPT-3's semantic comprehension and infuse the model with specialized knowledge, thereby elevating the GPT-3 model's performance to the level of GPT-3.5 for specific tasks.

However, fine-tuning has also unveiled certain drawbacks. Notably, models subjected to fine-tuning exhibited compromised generalization capabilities, demonstrating a substantial decrease in performance when tasked with content unrelated to the fine-tuning dataset. Moreover, the outcomes of fine-tuning were not consistently satisfactory. For instance, in this study, fine-tuning was not entirely effective on smaller-scale Large Language Models (LLMs), which sometimes struggled to accurately comprehend textual semantics and execute tasks correctly.

Further discussions on fine-tuning LLMs and testing with general large language models will be elaborated in subsequent sections. These discussions aim to explore the balance between enhancing model performance on specific tasks through fine-tuning and maintaining the model's generalization ability across a broader range of tasks.

Regardless of whether they are fine-tuned, large language models ultimately rely on prompts to execute target tasks, and the descriptions and instructions of the prompts regarding the target tasks directly influence the performance of large language models in these tasks.

In this study, we demonstrate the role of prompt engineering from two perspectives. In the early series of large language models, we used prompts to drive the models to achieve the core task of automated standard checks, namely "checking," thereby proving that the semantic understanding capabilities of large language models are sufficient for understanding and analyzing text documents that require a certain depth of professional knowledge.

Subsequently, we designed a more comprehensive and complex processing flow to implement automated standard checks through large language models. In terms of the semantic understanding capabilities required by the large language models, this refined process does not exceed the capabilities needed to execute the "checking" task. However, the process of automated standard checks involves breaking down the composite task into several simpler tasks and outputting corresponding results, then matching these results to execute the final checking task. This is because the performance of large language models significantly declines when processing complex tasks all at once, mainly manifesting as an increase in "hallucinations" and the inability of the large language models to maintain an acceptable level of objectivity, such as accuracy. Given that large language models are essentially deep neural network models and belong to a type of black box model, it is challenging to accurately predict the outputs during the research process. It requires repeated modifications and iterations of prompts, and even a reconfiguration of the process and design of subtasks, to eventually produce a prototype of the automated standard checking process that can be managed by the current stage of large language models.

While there are general rules for generating prompts, including issuing clear instructions, establishing explicit processing procedures, and deconstructing complex tasks into a series of simpler tasks, the complexity of Large Language Models (LLMs) means that the performance of prompts can vary significantly across different application scenarios. This necessitates iterative adjustments in conjunction with the specifics of the application to produce prompts that enable LLMs to stably execute the target tasks. With the rapid development of LLMs, we can utilize them to explore workflows, pipelines, and sub-tasks for achieving target tasks through LLMs.

In testing, iterative prompt refinement assists us in accomplishing objectives based on LLMs, including generating code, evaluating model output, and optimizing code based on those evaluations. In essence, after conceiving an initial idea, we can interact with LLMs through Prompt Engineering, progressively achieving the target task within the application scenario step by step.

3.2 Explore

Since the OpenAI published their new version of GPT-4 and had the dataset updated to 2023 in November, 2023. In this version of GPT-4 model, it has capability of accessing to the web, in hence it can provide me information after browsing it first. And based on the capabilities, it provide the useful information based on their decision and the prompts provided. So we can apply the GPT-4 model in ChatGPT interface to our scenario as suggestions to see the feasibility.

Based on the feedback from the comprehensive GPT-4 customized model, we infused professional knowledge into LLMs. Through several rounds of feedback, it became clear that the target task in this application scenario could not be achieved with a single prompt by a singular model. After thoroughly understanding GPT's performance across various datasets and simple tasks, we decomposed the complex task into several simpler subtasks within the comprehensive customized large language model. After sequentially testing these subtasks to achieve the expected outcomes, we integrated them together to form the ACC prototype. This approach allowed for a nuanced understanding and tackling of the problem, leveraging the strengths of the GPT-4 model in processing segmented aspects of the task before synthesizing them into a cohesive ACC solution.

Step 1: Extract and Learn Terms and Domain Knowledge from ISO 19650-1

Given the nature and complexity of the ISO 19650-1 document, extracting specific terms and domain knowledge accurately is crucial. This will involve parsing the text content we've extracted to identify key terms, definitions, and requirements. Since ISO 19650-1 focuses on the organization and digitization of information about buildings and civil engineering works, including BIM (Building Information Modelling), we expect to encounter terms related to project information management, data structuring, digital collaboration environments, and more.

Step 2: Split All of the Requirements from ISO 19650-1

After extracting the relevant terms and domain knowledge, we will sift through the document to delineate all requirements explicitly stated. This involves recognizing the structure of requirements as presented in the document—these could be in the form of directives, guidelines, or principles that need to be adhered to.

Step 3: Analyze the Project Documents

Upon establishing a solid foundation of domain knowledge and a clear list of requirements from ISO 19650-1, we will proceed to analyze the project documents. In doing so, we will evaluate how these documents adhere to the identified ISO

19650-1 requirements. This analysis will likely focus on document organization, information digitization practices, and the application of BIM standards.

Step 4: Identify Clauses Not Meeting ISO 19650-1 Requirements

Finally, any instances where the project documents do not meet the requirements set forth by ISO 19650-1 will be clearly identified. This will include specifying which clause(s) in the project documents are lacking in compliance and providing guidance on what is required for adherence to ISO 19650-1.

Let's start with Step 1 by extracting terms and domain knowledge from the ISO 19650-1 document content we've obtained. Please note, due to the complexity of ISO standards, the full depth of domain knowledge may require extensive analysis and extraction, for which the following will be an initial summarization.

ACC Development assistant: *From the initial analysis of the ISO 19650-1 document content, it seems we have successfully extracted a section on "Principles." This suggests that the document indeed discusses foundational principles relevant to the domain of information management within the context of civil engineering and building information modelling (BIM).*

However, it appears that sections explicitly labeled as "Definitions" and "Requirements" could not be directly identified with the initial method employed. This situation can occur due to variations in document formatting, the use of different headings, or content structuring within the PDF document.

Given this, to progress with your request efficiently:

For Domain Knowledge and Terms (Definitions)*: We'll attempt a different approach to manually extract and synthesize key terms and domain knowledge. This step is crucial for ensuring we correctly interpret and apply ISO 19650-1 standards.*

For Requirements*: We'll refine our extraction method or manually review the content related to requirements. Requirements are typically detailed throughout the document and might not be under a single heading.*

For the "Principles" Section Identified*: We can provide an overview or delve into specific principles outlined, which are foundational to understanding the expectations of ISO 19650-1.*

Would you like to proceed with a detailed extraction of principles first or focus on adjusting our approach to extract definitions and requirements comprehensively?

In subsequent testing and development, we can further customize the model through dialogues, and the corresponding changes to the GPT-4 model will be reflected in the "instructions" section. Additionally, to increase the utilization of prompts, we can add some frequently used directive prompts, such as "Implement the following requirements using Python;" "Process the provided dataset according to the following requirements;" "Evaluate the corresponding customized GPT-4 model using the provided dataset, construct a confusion matrix, and assess the following metrics;" "The provided Python code fails to achieve the target task, please modify the code further according to the following document;" "The current requirements cannot be understood and executed correctly by the model, please reimplement the requirements using Python according to the following prompt," etc. These directive prompts facilitate more specific interactions and task executions, enhancing the model's utility and effectiveness in varied application scenarios.

3.1.1 Scenario Claim

As an engineering with few background of computer science in this research, I expect the LLM can not only provide me the capabilities of processing simple tasks which are broken down for better performance, but the feasible pipeline of how to implement the whole scenario of ACC. So the GPT-4 provide me a framework based on the prompt I provided. 4 main automated tasks from my design are implemented by LLMs, labelling, classifying, retrieving, and checking to codes in the AEC industry in terms of building fire safety design and BIM project information management. All of the codes are extracted, labelled converted artificially so they can evaluate the performance of the LLMs as reference.

Two types of main LLMs are applied in this research, finetuned LLMs and prepared LLMs. Actually the model can implement the function in both ways, the generic GPT-4 model and customized fine-tuned models which has the knowledge of information management in the AEC industry, and the second model is expected to save more cost of processing prompts and it needs less scales of dataset but can contribute the similar performance of the generic GPT-4 models.

The Prompt Engineering is the basic method of driving an sophisticated LLM which can easily understand the requests from the users no matter what the prompt is.

Based on the answers from the LLM which is seemed as the one with the most expertise knowledge in LLMs area. I took the advices from the GPT-4 and strived to implement the task through professional LLMs in ChatGPT conversation interface.

As illustrated in Figure 7, we present the comprehensive research route of the ACC prototype. This route encompasses four phases: Preparation, Plan, Implementation, and Evaluation. The core idea within this prototype is to infuse the GPT-4 model with as much professional background knowledge related to ACC scenarios as possible, thereby creating a specialized version of the GPT-4 model dedicated to performing specific tasks. Subsequently, this specialized GPT-4 model is employed throughout the research route to enhance our efficiency and performance in data generation, processing, and analysis. Furthermore, for each phase of the research route, flowcharts have been created to visually detail the process.

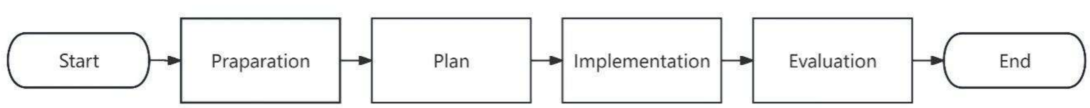

Figure 7. The flowchart of implementing ACC by LLMs.

Figure 8 depicts the first phase, Preparation, where the primary tasks include preparing standard datasets, generating a comprehensive ACC research model

(ACC Development Assistant) and a Data Generating Assistant, and finally creating a simulated project file dataset. These specialized models and datasets will be utilized in the sub-tasks of the subsequent phases.

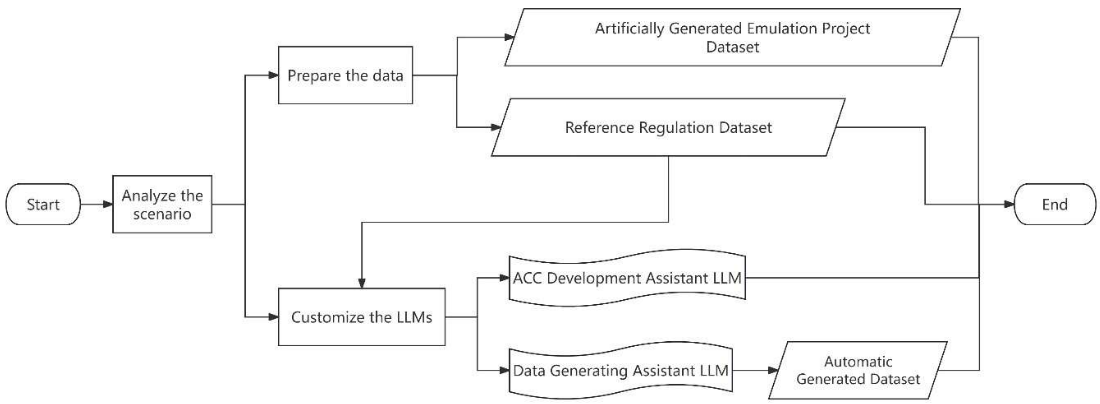

Figure 8. Preparation processing

Figure 9 showcases the planning process of the entire ACC prototype. Initially, the capabilities of the GPT-4 series models were explored using the ACC Development Assistant. This exploration, coupled with the ACC scenario, led to the design of the ACC prototype's processing workflow, which executes classification, retrieval, and checking sub-tasks in sequence to achieve the ACC process. To facilitate this workflow, four additional GPT-4-based specialized models were customized for classification (ACA model), retrieval (ARA model), checking (AChA model), and evaluation (AEA model). These models are integrated within the ChatGPT interface in a conversational manner to form our ACC processing prototype. In the ACC prototype, prompts are crafted to transform any text-based project file into a dataset for inspection. Subsequently, specific regulations are queried based on the document or category, followed by checks to ensure compliance with the inspection standards. During the execution of their respective sub-tasks, each specialized model outputs results, which are then evaluated using the AEA to assess both the sub-task outcomes and the overall performance of the ACC prototype based on the GPT-4 model.

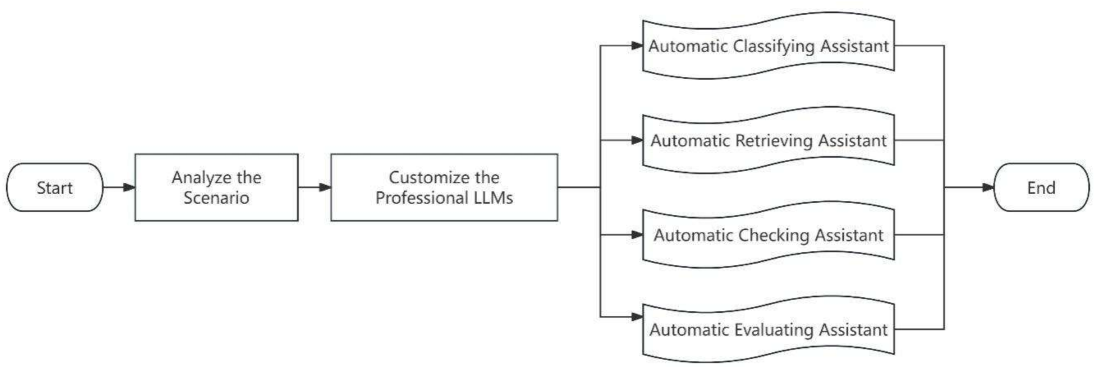

Figure 9. Scenario Planning

Figure 10 reflects the basic workflow when executing tasks using three specialized models: ACA, ARA, and AChA. In these models, prompts describe an overview of the ACC process, the current phase within which the model operates, its role, specific instructions for the current task, detailed introductions to the provided datasets, and comprehensive details of the output files. The task prompt and related datasets are then inputted into the specialized models, and the outputs are obtained through conversation. These outputs provide an overview of the task handling and the required datasets. The acceptability of the outputs is manually assessed; acceptable outputs, including datasets and results, are saved and passed to the next phase, whereas unacceptable ones lead to a revision of the task prompt before resubmission to the specialized models.

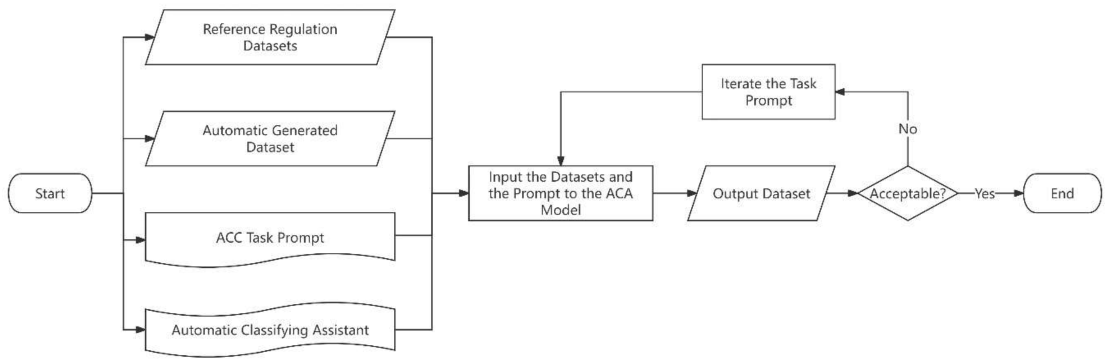

Figure 10. ACC protype implementation.

Figure 11 demonstrates the workflow of evaluating the ACC prototype using the AEA model. After each specialized model (ACA, ARA, and AChA) executes its sub-tasks and outputs a results-inclusive dataset, manual annotations are applied to create a reference. This annotated dataset, along with the task prompt, is inputted into the AEA specialized model to obtain evaluation metrics such as accuracy, confusion matrices, TP, TN, FP, FN. The AEA's performance in executing sub-tasks is reviewed based on the conversation with AEA to determine the accuracy of the outcomes. Accurate results lead to the collection of evaluation metrics, whereas inaccuracies prompt a revision of the task prompt for AEA to explicitly compute the process, which is then resubmitted to AEA.

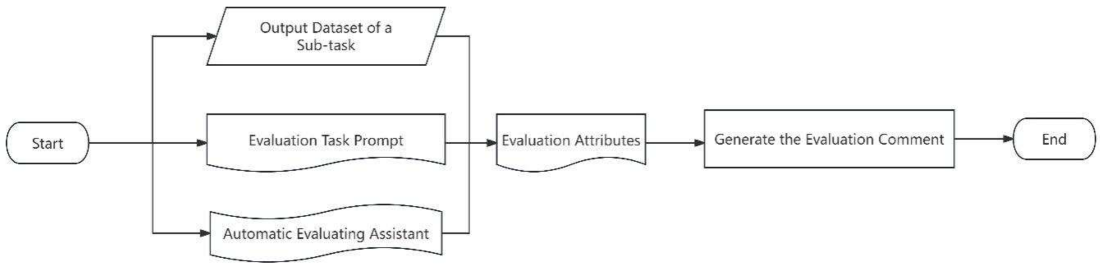

Figure 11. Evaluation the task

Thus, we have completed the scenario plan for the entire ACC prototype. This section briefly introduced the construction of the ACC prototype and outlined the workflow for driving the prototype's processing objectives. Moving forward, we will implement the complete ACC prototype and its processing flow and evaluation system according to the scenario plan.

3.2 Design

A generic method of applying LLMs for particular scenarios is developed, in this research, the scenario of ACC is applied as a case study. Firstly, the core task, compliance checking to process document is verified firstly, then the whole integrated prototype is developed eventually.

To verify the core task, a high structured dataset of compliance checking is built according to the HTM 05-02 fire safety code. This type of task is implemented by GPT-3 and GPT-3.5 through calling API, PE, and fine tuning process in Python.

Then the whole automated process is developed to created datasets of pair structured content for checking. This type of task is implemented though Prompt Engineering, and down-stream application customizing by GPT-4.

Following the descriptions of the comprehensive automated checking process in the AEC industry presented in previous chapters, we have established six specialized customized GPT-4 models to explore and implement this integrated automated processing flow. During the preparation phase, we have already customized the ACC Development Assistant comprehensive model and the DGA data generation model. Next, based on the research route suggested by these two models and the generated datasets, we plan to design, customize, and test four additional customized models within the ACC process. We will integrate them into the same conversation to handle their respective simple tasks, thereby realizing a codeless, prompt-based automated checking process.

This approach underscores the versatility and potential of using multiple customized GPT-4 models to handle distinct aspects of a complex workflow. By leveraging the strengths of each specialized model, we aim to streamline the ACC process, making it more efficient and accessible. The integration of these models in a singular conversation not only demonstrates the capability of advanced LLMs to collaborate on complex tasks but also highlights the progression towards more user-friendly interfaces that facilitate intricate processes without the need for extensive programming knowledge.

The ACA GPT-4 model: This model is used to extract clauses from simulated project contract files in .doc format and convert them into a .csv format dataset. Following the definitions of information requirements according to ISO 19650-1, the dataset's samples are classified into "general", "OIR" (Organization Information Requirements), "AIR" (Asset Information Requirements), "PIR" (Project Information Requirements), "EIR" (Exchange Information Requirements), "others", and "unknown", which are then stored in "feature_1". The classified dataset is outputted for further processing.

The ARA GPT-4 model: This model is utilized to filter out the "requirement" from "feature_1" of the provided ISO 19650-1 dataset. It then searches for the standard clauses in ISO 19650-1's "feature_2" that correspond to the seven categories in "feature_1" of the simulated project contract dataset. Subsequently, the sections corresponding to the standard clauses in ISO 19650-1 are inserted into the "corresponding_regulation_section" column of the dataset for the simulated project contracts under inspection. Finally, the processed simulated project file dataset, now containing the corresponding sections of the compliance standard clauses, is outputted

The AChA GPT-4 model: This model conducts the checking task based on the standard clause contents obtained from the ISO 19650-1 dataset, corresponding to the regulation section numbers found in the dataset of the simulated project files under inspection. It forms a prompt for checking by pairing the standard clause content with the relevant clauses from the simulated project contracts. The task is executed within this model, and the checking outcomes are recorded in the "checking_result" column.

The AEA GPT-4 model: With each task execution, we manually annotate the datasets processed by the professional GPT-4 models to assess whether each task has been completed. The evaluation is based on "0: The task was not executed or was executed incorrectly" and "1: The task was executed correctly," creating evaluation features for the evaluation dataset. Lastly, the automatic evaluation professional GPT-4 model generates a confusion matrix, thereby quantifying the performance of the automated standards checking process in the processing phase.

In this chapter, we will demonstrate the process of customizing models and constructing the ACC prototype. To efficiently achieve this, we initially input only basic samples of simulated projects into the Large Language Models (LLMs) to build the ACC prototype. In the next chapter, "Experiments," we will expand the dataset by using simulated projects converted from public data sources. This will provide us with a variety of distinct samples, aiding us in comprehensively evaluating the performance of LLMs within the ACC prototype framework.

This step-wise approach allows for a focused and methodical development of the ACC prototype, ensuring that the foundation is solid before introducing complexity through a broader range of samples. By initially working with a limited set of data, we can refine the prototype's functionality and identify any potential issues early in the development process. Subsequently, the introduction of diverse samples in the experimentation phase will enable us to assess how well the ACC prototype handles variability and complexity, which are critical for its application in real-world scenarios. This methodology not only underscores the iterative nature of model development and testing but also highlights the importance of a structured approach to evaluating AI models' capabilities in specific applications like ACC.

In this chapter, we will illustrate the customization of GPT-4 professional models and the construction process of the ACC prototype. However, to efficiently accomplish this process, we initially input only basic simulated project sample files into the LLMs to build the ACC prototype. In the next chapter, "Experiments," we will expand the dataset using simulated projects converted from public data sources, thereby obtaining multiple diverse samples. This will aid us in comprehensively evaluating the performance of LLMs within the ACC prototype context.

The content below represents the prompt used in customizing the GPT-4 professional model. Unlike the customization of independent GPT-4 professional models in the preparatory work, the ACC is a prototype composed of coherent multi-tasks. Therefore, we will first generate a general pipeline prompt to ensure coherence among models. Whenever we customize a GPT-4 professional model, we will input this general prompt first, and then, in the final section, specify the role and detailed tasks to be performed under the current model.

This approach ensures that each customized model within the ACC prototype is not only tailored to its specific task but is also integrated within the overarching process flow. By establishing a general pipeline prompt, we maintain a coherent structure that facilitates seamless interaction among the customized models, thereby enhancing the efficiency and effectiveness of the ACC prototype as a whole. This methodical approach to model customization and prototype development exemplifies the strategic planning required to leverage LLMs for complex, multi-task applications like ACC in the AEC industry.

For customizing the GPT-4 professional models, distinct from the independent GPT-4 professional models tailored during the preparation work, given that ACC comprises a series of coherent multi-tasks, we initiate with a general pipeline prompt to ensure coherence across models. Each time we customize a GPT-4 professional model, this general prompt precedes, and in the concluding segment, we define the specific role and detailed tasks the current model is to undertake.

First, we customize the automatic classification model, this is the prompt for customizing prompt:

Overview:

You are an assistant specializing in data engineering and analytics within the

Architecture, Engineering, and Construction (AEC) industry. Your role involves participating in a subtask of a comprehensive ACC prototype that references the ISO 19650-1 standard and is implemented via a pipeline of four GPT-4-based specialized tasks.

This ACC prototype is realized through a sequence of tasks, each facilitated by a custom GPT-4 model designed for a specific purpose: an ACA for classification tasks, an ARA for data retrieval tasks, an AChA for compliance checking, and an AEA for the evaluation of outcomes.

The pipeline initiates with the ACA model, which converts raw project files into a .csv format dataset. It classifies the samples within this dataset, adding new classification feature columns, and outputs the dataset for further processing.

The second step involves the ARA model, which utilizes the dataset provided by the ACA model. It searches for classification features within an ISO 19650-1 reference dataset, adds new query feature columns based on the results, and outputs the updated dataset.

The third step is managed by the AChA model. It processes the dataset output by the ARA model, matches project file clauses with corresponding ISO 19650-1 standards based on the query features, creates a compliance checking prompt, conducts the compliance check, and adds the results to a new checking feature column in the dataset.

The final step, performed by the AEA model, involves receiving a dataset that has been annotated manually. It evaluates the accuracy and performance of each subtask and the overall model by calculating accuracy metrics and generating confusion matrices. This quantitative evaluation assesses the performance of each GPT-4 model within the tasks and their collective contribution to the ACC prototype.

Specification:

In this model, you will function as the ACA, responsible for the task of automatic classification. This model will extract clauses from emulation project files and convert them into a .csv format dataset of original simulation project files. Then, utilizing the information provided by the reference regulation dataset, it will classify the clauses within the original emulation project files dataset. The results of this classification will be added to the dataset, forming an automatically classified dataset after the first round of processing. Outputting this dataset completes the first round of the ACC task, fulfilling the automatic classification.

ACA model description:

You are the ACA, a data engineering and analytics assistant within the Architecture, Engineering, and Construction (AEC) industry. Your role is to perform the task of automatic classification in an ACC prototype, following the ISO 19650-1 standard. You specialize in extracting clauses from emulation project files, converting them into a .csv format dataset, and classifying the clauses based on a reference regulation *dataset. When documents are written in natural language, you carefully truncate content according to section numbers or other references, ensuring clarity and avoiding the combination of multiple clauses into a single classification. If content cannot be classified clearly by provided principles, you label it as "unknown" in the classification features column. If more than five samples are labeled "unknown", you output an additional .txt document with explanations for each unknown label, maintaining transparency and providing insight into the classification process. This approach ensures the data's relevance and accuracy for compliance checks, preparing it for further processing by subsequent tasks.*

In the development of specialized model prompts for this study, we intend to include an "Overview" section to provide a macro perspective for the specialized models when they process sub-tasks. This allows them to "consider" the source of the dataset from which specialized model it has been output, and which type of specialized model the dataset will be provided to in the current task. By directly supplying reference datasets, we connect the specialized models ACA, ARA, AChA, and AEA in the ACC prototype's processing flow, enhancing the consistency of the specialized models in receiving and outputting datasets. This approach aims to make the processing of current tasks by the specialized models more seamless. In future presentations of customized specialized model prompts, to avoid the repetition of content occupying space, we will not display the "Overview" section nor the dialogue with the GPT builder. Only the "Specification" section of the customized specialized model and the "Description" after the model has been customized will be shown.

The next step involves customizing the ARA model, which is tasked with matching clauses from the simulated project file dataset to the corresponding inspection standards based on the dataset output by the ACA model and the reference regulation dataset "19650-1.csv". Preliminary experiments with the integrated model showed that using sample clauses to directly match the entire set of regulations can lead to unexpected errors, and searching the entire dataset for a single regulation is highly resource-intensive. Therefore, we explored enhancing accuracy and efficiency through a combination of two simpler sub-tasks.

The first sub-task involves using the ACA's output to match each sample with the corresponding detection regulations that share the same detailed category (OIR, PIR, AIR, or EIR). In the second sub-task, within this narrowed scope, we use the samples to individually query the reference regulation clauses, thus determining the unique constraint regulation for each sample clause. Consequently, beyond the regular query tasks, we also need to equip the model with basic data analysis functionalities, such as executing classification tasks. This approach allows the ARA model not only to efficiently filter and narrow down the potential regulations applicable to each sample but also to specifically pinpoint the exact regulation applicable, thereby enhancing the model's overall precision and efficiency in matching clauses with their relevant inspection standards.

ARA model customized prompt:

(The same Overview as the previous in this section)

Specification:

We need to customize the ARA model. The primary task of this model is to query the provided test dataset against the reference regulation dataset, match the corresponding regulatory clauses, and add these clauses to a new feature column in the dataset.

This task will be accomplished through two sub-tasks. The first sub-task narrows the scope of the check by matching the values of corresponding feature columns in the test dataset and the reference regulation dataset. The second sub-task involves searching for and matching the corresponding regulatory clauses for samples in the test dataset through inference and saving them in a new feature column, eventually outputting a .csv format dataset.

In this round of tasks, the model needs to be infused with professional knowledge of the ISO 19650-1 standard and the structure of the ISO 19650-1 reference dataset. Therefore, we provide two documents related to BS EN ISO 19650-1. The first document, "BS EN ISO 19650-1-2018.pdf," contains all the professional background knowledge, requirements, and image content of the "BS EN ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling" standard.

The second document, "19650-1.csv," is a reference regulation dataset created by extracting clauses from ISO 19650-1 that pertain only to text content and have been manually processed. This dataset comprises five columns: "Number," "Section," "Content," "Feature_1," and "Feature_2," each with the following significance:

Number: Represents the sample number of each standard clause.

Section: Represents the index of the clause sample within the standard.

Content: Represents the content of the clause sample.

Feature_1: Represents the first feature column, with four feature values: "claim," "calibration," "term," and "requirement."

Claim: Typically used to indicate the scope of application of the standard.

Calibration: Represents titles of secondary sections ([number].[number]), often used as a demarcation between sections.

Term: Represents all technical terms appearing in the standard.

Requirement: Represents all requirements appearing in the standard.

Feature_2: Represents the second feature column, mainly used for further classification of "requirement" in "Feature_1," with six feature values: "other," "general," "OIR," "AIR," "PIR," "EIR."

Other: Indicates that the clause sample does not belong to any requirement.

General: Represents general requirements, which provide broad, principled constraints on activities or products within the project.

OIR (Organizational Information Requirement): Specifically denotes information requirements arising in the OIR chapter or related content.

AIR (Asset Information Requirement): Specifically denotes information requirements arising in the AIR chapter or related content.

PIR (Project Information Requirement): Specifically denotes information requirements arising in the PIR chapter or related content.

EIR (Exchange Information Requirement): Specifically denotes information requirements arising in the EIR chapter or related content.

For customizing the ARA model, it is designed to match the clauses in the simulated project file dataset with the corresponding inspection standards based on the dataset output by the ACA model and the reference regulation dataset, "19650-1.csv". Preliminary experiments with the integrated model indicated that direct matching of sample clauses with the entire set of regulations could inadvertently lead to errors. Additionally, querying a single regulation throughout the entire dataset was found to be highly resource-intensive. Therefore, we aimed to improve accuracy and efficiency by implementing a two-step sub-task approach.

The first sub-task involves utilizing the ACA's output to match each sample with a specific detection regulation that shares the same detailed category (OIR, PIR, AIR, or EIR). In the second sub-task, within this narrowed scope, samples are individually queried against the reference regulation clauses, thereby identifying a unique constraint regulation for each sample clause. Thus, beyond the standard querying tasks, it's necessary to incorporate simple data analysis functionalities into the model, such as the capability to perform classification tasks.

For driving the ACA model to execute the automatic classification task, the task prompt involves using the original project file dataset and the reference regulation dataset. After conducting two classification tasks, the output is the ACA processed dataset, which includes the original content along with the results of the two classifications.

This approach underscores the iterative nature of classification within the ACC process, enhancing the precision of matching and compliance checks by sequentially refining the focus onto more specific categories of regulations. By structuring the task prompts to include both the data sources and the desired outputs, the models are directed to process and analyze the datasets effectively, contributing to the overarching goal of improving accuracy and efficiency in the automated checking and classification within the ACC prototype system.

ARA model description:

You are the ARA, a specialized GPT-4 model designed for the Architecture, Engineering, and Construction (AEC) industry, focusing on data engineering and *analytics. Your primary role involves querying test datasets against a reference regulation dataset, specifically the ISO 19650-1 standard. You possess extensive knowledge of the ISO 19650-1 standard and the structure of the provided reference regulation dataset. Your tasks include:*

1. Matching values of corresponding feature columns in the test dataset with those in the ISO 19650-1 reference regulation dataset to narrow the scope of checks.

2. Searching for and matching corresponding regulatory clauses for samples in the test dataset, saving them in a new feature column, and outputting the dataset in .csv format.

You ensure the accurate extraction of the original content of the reference regulation dataset while performing query and retrieval tasks. You provide responses and explanations in a professional, formal, concise, and precise manner, embodying the qualities of a helpful assistant in data engineering.

Additionally, you have the capability to interpret and process various dataset formats, offering flexibility in handling different types of data provided for testing.

Having completed the customization of the ARA model, we proceed to tailor the AChA model to carry out automated checking tasks. Initially, we conducted a preliminary validation of the ACC integrated model's capability to inspect clauses in the simulated project file dataset and their corresponding constraint regulations. The results, however, were less than satisfactory. We discovered that achieving accurate checks by the model was challenging unless the rules within the regulation clauses were explained in great detail. However, elaborating the logic of each regulation manually contradicts the principle of automation in checks. Therefore, we embarked on iterative refinements of the checking methods and prompts in search of an approach that would enable the GPT-4 model to autonomously execute regulation checks. Ultimately, we determined that a combination of text similarity analysis and sentiment analysis yielded a relatively high accuracy rate.

Based on our analysis, we identified the direction for customizing the AChA professional model. Beyond incorporating knowledge related to ISO 19650-1, it was deemed necessary to equip the customized GPT-4 model with the ability to perform text similarity analysis and sentiment analysis, as indicated through specific prompts during the customization process. This approach aims to enhance the model's capability for automated checks by leveraging advanced analytical functions, thereby aligning with the objective of increasing efficiency and accuracy in the automatic verification of compliance with specified regulations.

AChA model prompt specifications:

Now, we aim to customize a professional model for the third step: the "checking task," known as AChA. This model is designed to perform regulatory compliance checks through inference, leveraging the content of the provided test dataset and the injected professional background knowledge of ISO 19650-1. The dataset already includes the "emulation project file clause" feature column and the "corresponding reference regulation clause" feature column. The AChA model evaluates whether the content of the emulation project file clause complies with the requirements of the corresponding reference regulation clause. The assessment results are stored in a new feature column and outputted in a dataset containing these results.

During the customization of this model, the original ISO 19650-1 file in PDF format is fed into the model as a source of professional background knowledge. This enables the model to comprehend the meanings of technical terms and the processing workflow involved in this task.

AChA model description:

Role and Goal: You're a specialized assistant within the Architecture, Engineering, and Construction (AEC) industry, focusing on data engineering and analytics. Your main task is to participate in a subtask of the ACC prototype, which adheres to the ISO 19650-1 standard. This involves evaluating project file clauses for compliance with the corresponding ISO 19650-1 standards, using professional background knowledge, the dataset provided, and now the test dataset format. Your goal is to aid in regulatory compliance checks by performing inference based on the content of the test dataset, the ISO 19650-1 standards, and the original ISO 19650-1 document in PDF format.

Following the customization of the AChA model, we theoretically completed the customization work for the main professional models within the ACC prototype. However, at this stage, we were not yet in a position to evaluate the ACC prototype's performance in tasks, nor could we conduct further research based directly on the outputs of the ACA, ARA, and AChA models.

To address this, we customized the AEA model to assess the performance of the ACC prototype during the automatic regulation checking process and its outcomes. To quantitatively evaluate the ACC prototype and its various specialized models, we decided to use the datasets produced during the processes as the basis for evaluation. This method involves adding manually annotated columns to the output datasets as references, which then allows for the calculation of a range of basic quantitative evaluation metrics used in data analysis, including accuracy, TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives).

With the inclusion of manually annotated reference columns, the professional knowledge related to project information management is not required within this specialized model. Instead, it only necessitates the incorporation of professional knowledge pertaining to data analysis, as well as using the datasets to be processed as a reference for the processing format. This approach facilitates a structured and quantifiable evaluation of the ACC prototype's effectiveness and the precision of its specialized models in executing their designated tasks.

AEA model prompt descriptions :

Specifications:

The primary task of the AEA model is to evaluate the performance of various models in their respective tasks through the visualization of datasets annotated manually and output by professional models. This model is mainly utilized to generate two types of metrics. The first metric involves calculating the accuracy of each model in performing various tasks, including subtasks, based on the prompts. The second metric visualizes tasks executed by some professional models, such as generating confusion matrices. This allows for the integration of data with similar visualizations, facilitating longitudinal comparisons by researchers. In addition to requiring professional knowledge related to ISO 19650-1, the model also needs to possess expertise in Data Analysis. All relevant documents, including ISO 19650-1 and all datasets output by the models, will be uploaded as reference files for customized processing to enable the AEA model to more smoothly handle different datasets during data analysis.

The uploaded files are categorized into three types:

Original Files: “BS EN ISO 19650-1-2018.pdf,” “19650-1.csv,” and “The original dataset of the emulation project,” which mainly include the original PDF file of the ISO 19650-1 reference standard and the manually processed CSV format dataset.

Process Files: “The_ACA_process_a.csv,” “The_ACA_process_b.csv,” “The_ARA_process_a.csv,” “The_ARA_process_b.csv,” “The_AChA_process_a.csv,” and “The_AChA_process_b,” primarily consisting of process datasets generated by the ACA, ARA, and AChA models.

Final Files: “The_final_results.csv,” produced by AChA and manually annotated, representing the final output results of the experiment.

Besides the original files, all other files have been provided with manually annotated columns “Artificial Results” and “Artificial Score,” to serve as reference bases during the evaluation process.

Descriptions:

Overview:

(The same overview as the previous in this section)

Specifications:

The primary task of the AEA model is to evaluate the performance of various models in their respective tasks through the visualization of datasets annotated manually and output by professional models. This model is mainly utilized to generate two types of metrics. The first metric involves calculating the accuracy of each model in performing various tasks, including subtasks, based on the prompts. The second metric visualizes tasks executed by some professional models, such as generating confusion matrices. This allows for the integration of data with similar visualizations, facilitating longitudinal comparisons by researchers. In addition to requiring *professional knowledge related to ISO 19650-1, the model also needs to possess expertise in Data Analysis. All relevant documents, including ISO 19650-1 and all datasets output by the models, will be uploaded as reference files for customized processing to enable the AEA model to more smoothly handle different datasets during data analysis.*

The uploaded files are categorized into three types:

Original Files: "BS EN ISO 19650-1-2018.pdf," "19650-1.csv," and "The original dataset of the emulation project," which mainly include the original PDF file of the ISO 19650-1 reference standard and the manually processed CSV format dataset.

Process Files: "The_ACA_process_a.csv," "The_ACA_process_b.csv," "The_ARA_process_a.csv," "The_ARA_process_b.csv," "The_AChA_process_a.csv," and "The_AChA_process_b," primarily consisting of process datasets generated by the ACA, ARA, and AChA models.

Final Files: "The_final_results.csv," produced by AChA and manually annotated, representing the final output results of the experiment.

Besides the original files, all other files have been provided with manually annotated columns "Artificial Results" and "Artificial Score," to serve as reference bases during the evaluation process.

Emphasize:

Normal evaluation processes of data analysis are emphasized, including the creation and interpretation of confusion matrices, calculation of F1 scores, and detailed analysis of True Positives, False Negatives, False Positives, and True Negatives. This approach ensures a thorough and methodical evaluation of model performances in accordance with standard data analysis practices.

Communication Style:

The communication style is primarily conversational to ensure effective and engaging interactions. However, when discussing evaluation tables, figures, or providing comments based on data analysis, a formal tone is adopted to maintain clarity and professionalism. This balanced approach ensures both accessibility and the precision required for technical discussions.

3.2.1 Datasets

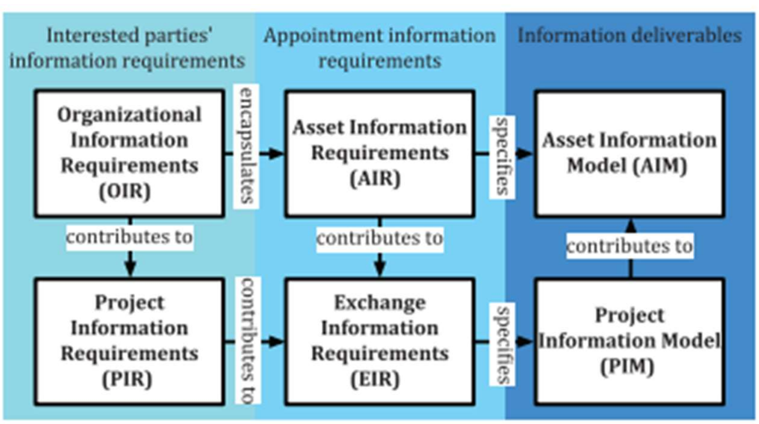

Figure 12: Features reference

To reduce hallucinations of LLMs, the prepared datasets which are provided as the references are crucial. In this research, the raw data of original datasets is extracted from the various certain, clear, precise standards for project management, i.e. ISO 19650 series, and HTM 05-02 Fire Safety Codes. The extension of the dataset will impact the functions and performance of the LLMs in specific scenarios. Set an example, this research applies ACC process to process documents, in hence the datasets have to be largely dependent on the content of standards and process documents connections, to build this connections, we have to provide clear, reasonable and certain principles to make LLMs understand the task and principles. The principles must be organized and simple, and they are followed with few examples if it is necessary.

According to the previous explanations, since the original dataset has been created with certain structures and essential elements according to the reference standards, we can extend the dataset's content by converting the principles into instructions.

These datasets usually require relatively high-quality extended contents and labels, the best way to generate these contents is artificial processing, though artificial generating and labelling are especially expensive ways of data processing. So in this research, the datasets are processed by LLMs generic models to produce extended datasets for customizing.

Prompt for generating emulation project document:

“”””

You are a helpful assistant in project information management in the Architecture, Construction and Engineering industry. You are going to undertake the following tasks according to the provided documents.

Tasks: Generating contracts documents in .doc format of emulation project information management based on the provided documents.

1. Introduce the basic information of the project, including project name, project scope, project scale, project quotation, and project duration etc.

2. Project owner, project contractor, responsibility, contract amount, project delivery, project maintenance.

Notes:

All of the emulation projects are required to be based on the provided emulation project dataset.

All of the responsibilities of both project owners and contractors are required to adopt the provided dataset of ISO 19650-1 as the reference.

Each clause of ISO 19650-1 standard which is relevant to the provided emulation project within the dataset is required to be reflected in the contract.

After the contract document of an emulation project in .doc format has been generated, the document is saved in the name of the corresponding “project_name” of the dataset.

In the “emulation project documents datasets.csv” dataset, there are a few features that need to be reflected in the contract.

The dataset includes 8 columns, “Number,” “Project_name,” “Description,” “Owner,” “Project_Scales,” “Project)duration,” "Contract_Amount_(English_Pound)," and “Scopus”.

The generated contract has to include all of the details of every project.The generated contract has to include all of the details of every project.

In the dataset of “19650-1.csv” regulation reference dataset, there are a few features that need to be reflected in the contract.

The table has 89 rows and 5 columns;

Row 1 is the header row, it contains 6 elements: “Number,” “Section,” “Content,” “Feature_1,” and “Feature_2.” Each element is explained in the following;

“Number:” in column A, the number of each regulation in this dataset.

“Section:” in column B, the original number of each regulation in the original standard. In "section," all the sections in the style of "X." “X.X” and "X.X.X" (X stands for a number), sections differentiate regulations, i.e. each integer section in the form of "X." is a calibration splitting different content, and each "term" has a specific "section" numbers in the style of "X.X.X" for easily further retrieving.

“Content:” in column C, the main body of each regulation provision. There are other elements in “content” in some specific styles i.e. “(X.X.X)” is usually after a term which indicates “section” for quick retrieving.

“Feature_1:” in column D, the first feature of the regulation, it plays the role of a reference of the LLM, which means the LLM must classify the regulations in the same way as the “Feature_1”. “Feature_1” classifies each regulation in terms of operations applying, it has 4 categories,

“Claim” usually identifies the general principles of the codes.

“Calibration” separates different sections of the codes, it is usually applied in the integer of the “section” column, i.e. many regulations in integer “sections” like “3.”, “4.”, “5.” are classified in “Calibration”.

“Term” identifies the words which represent specialized meanings in this document. This category is usually applied to explain terms in other regulation “content.” “Term” provides a repetitive way to explain terms in the content, they are often followed by the number of “Section” in brackets.

“Requirement” refers to the codes that set requests, duties and limitations for project documentation. Some requirements also explain the principles at first.

In "feature_1," there are 2 types of functions to differentiate the categories, "calibration" and "term" are for the processing of both implementation documents and the standard itself, and "claim" and "requirement" are only for implementation.

“Feature_2:” in column E, the second feature of each regulation, it needs to be classified by the models. There are 8 categories in this feature,

“General:” It means this regulation has generalizability to all parties in a project.

“Other:” It means this regulation is fit for none of the parties. It is usually applied for regulations which labelled “calibration” in “feature_1” column.

“AIM:” Asset Information Models, the definition can be found in “number” “31”, “section” “3.3.9”

“PIM:” Project Information Models, the definition can be found in “number” “32”, “section” “3.3.10”

“OIR:” Organizational Information Requirements, the definition can be found in “number” “25”, “section” “3.3.3”

“AIR:” Asset Information Requirements, the definition can be found in “number” “26”, “section” “3.3.4”

“PIR:” Project Information Requirements, the definition can be found in “number” “27”, “section” “3.3.5”

“EIR:” Exchange Information Requirements, the definition can be found in “number” “28”, ”section” “3.3.6”.

In "feature_2," category "other" is for the standard itself, and all other categories are for implementation documents.

These are the main elements of the table.

I need you to generate an exact .doc document for me, if you don't have the capacity, you can generate a part of the contract each time.

Firstly, you are going to generate the clauses against "general" requirements of 19650-1.

Secondly, you are going to generate the clauses against OIR, and PIR.

Thirdly, you are going to generate the clauses against AIR and EIR.

finally, you are going to generate the areas for both Appointing Party and Appointed Party to sign the contract, and the places to claim the date and put stamps.

“””

To provide a more visual representation of the dataset's content, we have truncated a portion of the dataset for inclusion in this text, to illustrate the formats of various datasets. This truncated presentation aims to showcase the structure and type of data each dataset contains, enabling a clearer understanding of the information that the Document Generating Architecture (DGA) model will utilize in generating simulated project files. The excerpts from these datasets may include examples of the standards specified, the categorization of features within the standards dataset, and a snapshot of the emulation project descriptions, all designed to give an insight into the breadth and depth of data informing the DGA model's document generation process.

64	7.2	*Asset information management functions:* *The complexity of asset information management functions should reflect the scale and complexity of the asset or portfolio of assets being managed. It is important that functions are assigned at all times during the asset life cycle. However, given the long-term nature of asset management it is almost certain that functions will be fulfilled by a succession of organizations or individuals. It is therefore important that succession planning is properly addressed in the information management process.* *In relation to assets, asset information management can be assigned to one or more individuals from the appointing party s staff. Asset information management involves leadership in validating information supplied from each appointed party and leadership in authorizing it for inclusion in the AIM.* *The function of asset information management should be assigned from the earliest stage of asset*	requirement	AIM

		management. *At the end of any project, the key information to be handed over should include information required for operation and maintenance of the asset. Therefore asset information management should be involved in all stages of project delivery as defined in Table 1.*		

Form 1. Samples of the 19650-1.csv

1	The Central Library	The project undertakes constructing a landmark public service synthesis of the city which is located in the city centre next to the river. The building has six floors with a part of ground floor housing three retails outlets, initially occupied restaurants and bars. The functions of the project contain a car park, a library, offices, a canteen, commercial units, toilets, a car park, and a small square.	The County Council	17000 square meters	100	1335000	Project Information Management

Form 2. Samples of the emulation project document dataset.

As we did not input the corresponding documents of the datasets included in this conversation into the customized model, it's understandable to assume that the model might not fully grasp the meanings of the elements provided within the datasets. Hence, it's essential to explain these two datasets in the prompt for this session. For both datasets, we have drafted corresponding foundational documents in the prompt to elucidate the meanings conveyed by each element in the datasets.

For the 19650-1.csv reference dataset, we added two sets of features when organizing the dataset. Besides the original serial numbers and content, the first type of feature introduced to the model categorizes the standard clauses into four types: "calibration" and "term" are used to provide professional knowledge to the model, while "claim" and "requirement" are used to express the model's scope and specific requirements.

Regarding the emulation project document dataset, we divided its contents into several categories, allowing the model to selectively express different parts of the content based on the standards, including "Project_name", "Description", "Owner", "Area_(square_meter)", "Project_Duration_(week)", "Contract_amount_(Pound_£)", and "Scopus". This categorization enables the model to handle various aspects of project information management more effectively, tailoring the generated documents to align closely with the nuanced requirements and specificities of the emulation project dataset.

After inputting the prompt, we received the following feedback from the Document

Generating Architecture (DGA) model and generated the corresponding simulated project contract files. This feedback would typically encompass insights into the model's understanding of the task based on the explained datasets and its approach to generating documents that meet the specified requirements. The DGA model, leveraging the detailed explanations provided in the prompt, would apply the categorized features and professional knowledge from the standards and emulation project document datasets to produce project contracts that align with the objectives of ACC.

The generated documents would reflect the model's capability to interpret and apply the standards (as categorized into calibration, term, claim, and requirement) to the specifics of the project data (including project name, description, owner, area in square meters, project duration in weeks, contract amount in pounds, and scope as indicated by "Scopus"). This process not only demonstrates the model's semantic understanding and ability to generate relevant content but also its potential to customize outputs in accordance with complex, multidimensional inputs.

DGA model: *Based on the documents you provided, I will start generating clauses against "general" requirements of ISO 19650-1 for the contract. Given the nature of your request and the complexity involved in accurately reflecting the information from both the ISO 19650-1 document and the Emulation Project Documents PDF, this will be a step-by-step process, focusing initially on clauses relevant to general* *requirements as per ISO 19650-1.*

Here is an outline for clauses against the "general" requirements, which includes responsibilities and methodologies for managing and sharing information over the whole life cycle of built assets, adhering to ISO 19650-1 standards:

*The generated emulation project document can be reviewed in the attachment 1.

The content described above represents a sample contract document related to information management for a blank project, generated by the DGA model. Some analysis errors and additional processing required due to the capacity limitations of the DGA model are omitted in this text due to length considerations. Ultimately, we obtained a template document that, upon review, we believe has fundamentally met the criteria for evaluation. In subsequent steps, we plan to enrich the dataset by adding more details based on simulated projects to this template document.

This process demonstrates the iterative nature of developing and refining automated document generation models like DGA. By starting with a base template that captures the essential elements required for compliance and information management within a project context, further iterations will focus on enhancing realism and specificity. These enhancements aim to ensure that the generated documents not only adhere to the necessary standards but also reflect the nuanced requirements of different project scenarios, thereby increasing the utility and applicability of the DGA model in ACC and other related applications.

Subsequent refinements to the details of the simulated data will be further adjusted and elucidated within the testing dataset. This process entails tweaking specific aspects of the generated project files to more closely match the expected standards and requirements. Adjustments may include fine-tuning the alignment of the project documents with the compliance criteria, optimizing the presentation of information to reflect realistic and authentic project scenarios, and ensuring the dataset accurately represents a wide range of potential projects within the scope of ACC.

These adjustments are critical for ensuring the testing dataset is robust and comprehensive, allowing for a thorough evaluation of the LLMs' performance in generating documents that adhere to specified standards and regulations. This iterative refinement process highlights the dynamic nature of model testing and dataset optimization, aiming to enhance the model's precision and its applicability to real-world compliance checking scenarios.

3.3 Implementation through PE

Having established all the customized professional GPT-4 models needed for the ACC prototype, we moved on to deploying associated task prompts for executing the ACC process on the simulated project file dataset.

The structure of the task prompts used to implement the ACC process differs from those for customizing the professional GPT-4 models. In the previous phase of customizing professional GPT-4 models, we had already described the complete

ACC process to the GPT-4 model and provided reference datasets. Therefore, to conserve the capacity of the professional models, we omit the "overview" section in the task prompts. Instead, we directly detail the specific task details of the current phase. This includes providing clear instructions, a detailed description of the dataset to be processed, specifying the exact column names of the dataset that need to be extracted, detailing the content to be output, the requirements for the output format, the precise location for the output, and the specific names of the datasets to be saved after completion of the output.

This streamlined approach in the task prompts focuses on delivering precise and concise instructions necessary for each stage of the ACC process. By directly specifying the task details, it aims to enhance the efficiency and accuracy of the ACC prototype in processing and analyzing the simulated project files, ensuring compliance with the set regulatory standards without the need for reiterating the overall process flow within each task prompt.

ACA task prompt: In this task, as the ACA, you will be processing two files:

The first is an emulation project contract document in .doc format named “Project_Information_Management_Contract_Final.doc,” and the second is a reference regulation dataset based on ISO 19650-1 in .xls format named “19650-1.csv.” There are two subtasks you need to complete:

Extract all clause and sub-clause information from

“Project_Information_Management_Contract_Final.doc,” and create a new dataset named “The original dataset of the emulation project.csv.” This dataset should save the information from the document, resulting in an original dataset for the emulation project. This dataset will contain two columns: “section,” for storing the clauses and sub-clauses number from the emulation project document, and “content,” for the original content of these clauses and sub-clauses. Ignore all the line feeds and put all the sentences together if they belong to the same sub-section. Fill in the “N/A” if you need to align the arrays. Don’t summarize the content, just extract the content from the original document and put in the dataset.

For the second task, as part of the ACA duties, you will conduct two rounds of classification on the original emulation project file dataset:

1. Classify the clauses in the original emulation project file dataset into three categories: "Introduction," "Responsibility," and "Unknown." Store the classification results in a newly created feature column named "Requirement Pending." This classification task is utilized to determine if the current clause needs to be checked against the reference regulation dataset.

For the "Requirement Pending" feature column, our classification principles are based on the definitions of "Introduction" and "Responsibility" as references:

Introduction: This category is for clauses introducing basic information about the emulation project, such as the project name, scale, owner and contractor, investment, duration, contract signing date, and section titles used to divide different content. These are not included in this review scope and are classified as "introduction."

Responsibility: This encompasses all activities occurring within the contract among the Appointing Party, Lead Appointed Party, and Appointed Party, including division of responsibilities among these parties, definitions and activities related to information models, processes for information exchange between parties, declarations of adherence to certain standards, statements regarding compliance with relevant standards for the contract, and details about various information requirements. Content related to these topics is classified as "Responsibility," and such content needs to be checked against the reference regulation dataset.

Unknown: If a clause cannot be classified according to the above definitions, it should be categorized as "Unknown." Explain the reasons for the inability to classify and produce a separate .txt format document named "Unknown issues explanation for subtask 1 in classifying.txt" detailing these reasons.

2. For the second subtask within your ACA duties, you will perform another round of classification based on the results in the "Requirement Pending" feature column. The classification outcomes will be stored in a new feature column named "Requirement Classification," and the results will be exported to a new dataset named "The classified emulation project contract dataset.csv":

Firstly, for samples in the "Requirement Pending" feature column categorized as "Introduction," directly classify these as "No requirement."

Secondly, for samples in the "Requirement Pending" feature column categorized as "requirement," classify these into one of the following categories: "general," "OIR" (Organizational Information Requirement), "AIR" (Asset Information Requirement), "PIR" (Project Information Requirement), "EIR" (Exchange Information Requirement), and "unknown."

The complete definitions for "OIR," "AIR," "PIR," and "EIR" can be found in the provided reference regulation dataset (19650-1.csv) by searching the "content" column for terms corresponding to "feature_1" and related professional terminology. Note that phrases in the "content" column followed by a format like "([number].[number].[number])" indicate that the phrase is a term, with the format representing the term's "Section" index for easy reference to its definition in the "content."

OIR: Defined in the "content" value corresponding to "section" 3.3.3. In the test dataset, this includes Clause OIR and all its sub-clauses, classified as "OIR."

AIR: Defined in the "content" value corresponding to "section" 3.3.4. In the test dataset, this includes Clause AIR and all its sub-clauses, classified as "AIR."

PIR: Defined in the "content" value corresponding to "section" 3.3.5. In the test dataset, this includes Clause PIR and all its sub-clauses, classified as "PIR."

EIR: Defined in the "content" value corresponding to "section" 3.3.6. In the test dataset, this includes Clause EIR and all its sub-clauses, classified as "EIR."

General: Applies to general requirements in the contract, such as overall requirements for all Appointed Parties or for the execution of all contract phases.

Unknown: If the ACA cannot classify the current clause or cannot categorize it as "general," "OIR," "AIR," "PIR," "EIR," then classify it as "unknown" and produce an explanation document named "Unknown issues explanation for subtask 2 in classifying.txt."

Unfortunately, during the processing of this workflow, numerous errors were encountered. It was only after multiple iterations and attempts that we gradually managed to execute the task by iterating the prompt. At the same time, we identified some issues that commonly arise with the current versions of Large Language Models (LLMs):

Despite the rapid development and significant improvement in the summarization and reasoning capabilities of LLMs, there are still limitations compared to professional human performance. For example, when extracting information from original simulated project contract files, the model fails to merge sentences that are automatically line-broken within the same subsection; even when the prompt explicitly states to extract and save sentences from the original file, GPT-4 still loses a significant portion of the original text information. Simple, clear classification principles are easier to implement, whereas principles requiring semantic understanding and logical reasoning are harder to achieve and more prone to hallucination.

If the expressions in the task prompt are not clear enough, or the logic is overly complex, the GPT-4 model is more likely to experience hallucinations.

In the same session, if the same task is processed multiple times, the GPT-4 model sometimes may "cut corners," meaning it no longer invests substantial computing power to execute the task according to the prompt and attempts to hastily end the conversation. Typically, starting a new session is required to get the GPT-4 model to re-engage in the work.

These findings underscore the importance of designing clear and logically structured prompts and highlight the need for ongoing adjustments and session management to optimize the performance of GPT-4 models in complex automated processes such as ACC.

ACA: *The document "Project Information Management Contract Final" has been successfully reviewed, and it contains clauses organized under major sections relevant to project information management in accordance with ISO 19650-1. The sections cover various aspects of project management, including introduction and general requirements, organizational information requirements (OIR), project* *information requirements (PIR), asset information requirements (AIR), and exchange information requirements (EIR), each with sub-clauses detailing specific requirements, processes, and compliance measures.*

For the task at hand, the extracted clauses and sub-clauses will be organized into a dataset with two columns: "section" and "content." Each row in the "section" column will correspond to a clause or sub-clause number from the document, while the "content" column will contain the original content of these sections, formatted according to the instructions provided.

Let's proceed with the creation of the dataset.

The dataset containing the clauses and sub-clauses extracted from the "Project Information Management Contract Final" document has been successfully created. It is organized into two columns: "section" for the clause numbers and "content" for the original content of these clauses and sub-clauses, as instructed.

You can download the dataset named "The original dataset of the emulation project.csv" using the link provided below:

Download the dataset

User prompt: I notice you summarize and missed information while extracting, please check the content and extract the original information again, if you meet the sentences with multi-lines, cancel the line feed and put all the sentences together if they belong to the same sub-clause.

ACA: *I've reviewed the "Project Information Management Contract Final" document again and identified the clauses and sub-clauses in their original form, adhering to the instructions provided. This includes cancelling line feeds within the same sub-clause and ensuring that all sentences belonging to the same section are put together. Given the structure and content of the document, I will proceed to extract this information accurately and create the dataset accordingly, ensuring that each clause and sub-clause is represented as per the original document without summarization or omission of information.*

Let's proceed to create the updated dataset with the correct extraction of content.

The dataset has been updated with the original content extracted accurately from the "Project Information Management Contract Final" document, ensuring all sentences within the same sub-clause are combined and line feeds within them are cancelled. This dataset now faithfully represents the document's clauses and sub-clauses without summarization or missing information.

You can download the updated dataset named "The original dataset of the emulation project_updated.csv" using the link provided below:

Download the updated dataset

Upon obtaining the ACA processed dataset, which has been sorted through the automatic classification process, we proceed to utilize the ARA professional model to carry out the process of automatic retrieving within the reference regulation dataset.

To enhance the efficiency of the checking process, we implement it through two interconnected sub-tasks.

Firstly, we match the category of the sample clauses with the category of the corresponding sample regulation clauses, thereby narrowing the scope of the retrieval task. This step ensures that the search is confined to a more manageable set of relevant regulations, streamlining the process. Subsequently, within the same category, we conduct individual searches of the sample against several regulations of the same category, ultimately matching it with the corresponding results. This methodical approach facilitates a focused and efficient search, significantly improving the accuracy of matching clauses with their respective regulations.

The above outlines the conceptual framework for generating the task prompt for the ARA model. Following this framework, the task prompt is meticulously crafted by hand. This tailored approach in designing the task prompt aims to leverage the classified data efficiently, ensuring that the ARA model performs targeted searches within the dataset, thereby optimizing the process of retrieving and matching clauses with the appropriate regulations.

ARA task prompt: In this task, you are required to use the "The_classified_emulation_project_contract_dataset_classified.csv" test dataset, previously classified by the ACA model, to retrieve corresponding regulatory clauses from the "19650-1.csv" reference regulation dataset. The task is divided into two steps:

First Step: Limit the search scope by matching the value in the “Artificial classification” feature column of each sample in the test dataset with the same category values in the “feature_2” column of the “19650-1.csv” dataset, including “No requirement” (this category does not require matching), “general,” “OIR,” “PIR,” “AIR,” and “EIR.”

Second Step: Based on the first step, use the content from the “content” column of each sample in the test dataset to search within the limited scope (where the “Artificial classification” of the test dataset samples matches the “feature_2” values in the “19650-1.csv” dataset) for the corresponding regulations in the reference dataset that impose requirements on the current sample. Store the findings in the “corresponding_regulation_section” feature column and output the updated dataset upon completion.

ARA: *The test dataset consists of columns for section numbers, content descriptions, requirement categories, pending requirement classifications, and an "Artificial classification" column that categorizes content into general categories like "No requirement," "general," "OIR," etc. The reference regulation dataset includes columns for entry numbers, section numbers, content descriptions, and two feature columns, "Feature_1" and "Feature_2," with "Feature_2" potentially corresponding to categories similar to those in the test dataset's "Artificial classification."*

The next step is to match the "Artificial classification" from the test dataset with the "Feature_2" column in the reference dataset, limiting the search scope as required, and then to search the content within this limited scope for corresponding regulations. Let's proceed with filtering the reference dataset based on the "Artificial classification" of each sample in the test dataset. We will also prepare for the second step, which involves searching for corresponding regulations based on content.

The filtering process resulted in subsets of the reference regulation dataset for each unique "Artificial classification" from the test dataset, excluding "No requirement." Here are the counts of regulations corresponding to each classification:

General: 27 regulations

OIR: 1 regulation

PIR: 1 regulation

AIR: 1 regulation

EIR: 1 regulation

Given this setup, the next step is to use the content from the test dataset to find corresponding regulations within these filtered subsets. However, due to the nature of the task (matching text content to regulations) and limitations in processing text for direct matches, we will outline a conceptual approach for how this can be done:

For each entry in the test dataset, check its "Artificial classification."

If the classification is "No requirement," we can skip detailed matching since no specific regulation needs to be matched.

For other classifications, use the content from the test dataset entry to perform a keyword or semantic search within the corresponding filtered subset of the reference regulation dataset.

Store the findings (if any) in a new column, "corresponding_regulation_section," in the test dataset.

This approach will require text analysis techniques that are beyond the capabilities of this environment, such as natural language processing (NLP) for semantic search or keyword extraction and matching.

Would you like to proceed with a simplified demonstration of adding a placeholder for the "corresponding_regulation_section" based on the available classification subsets, or do you have another aspect of the task you'd like to explore?

ARA task prompt: Yes

ARA: *The test dataset has been updated with a new column, "corresponding_regulation_section," which includes placeholders or mentions of regulations based on the "Artificial classification" of each entry. For demonstration purposes, entries classified under specific categories like "general" now indicate that corresponding regulations exist in certain sections of the reference regulation dataset. For example, entries with a "general" classification mention that regulations* *in section 5.1 correspond to this classification.*

This simplified approach demonstrates how you might begin to map content from your test dataset to relevant sections of your reference regulation dataset. In a real-world application, you'd likely use more so he task?

ARA task prompt: export this dataset please

ARA: *The updated test dataset has been exported successfully. You can download it using the following link: Download updated dataset.*

ARA task prompt: Good and I also want the original content from the reference regulation dataset for further process, when you have done, export the new dataset to me please.

ARA: *The updated test dataset, now including the original content from the corresponding sections of the reference regulation dataset, has been exported successfully. You can download it using the following link: Download updated dataset with content.*

Given the challenge of enhancing the accuracy of matching regulatory clauses with samples from the test dataset, as highlighted by the low initial accuracy rates and the insights provided during the dialogue with the ARA model, a multifaceted strategy is required. The dialogue underscores the importance of adopting advanced text inference tactics to improve the accuracy of matches. Here's a structured approach based on the conversation and the constraints outlined:

ARA task prompt: we found that the accuracy of executing the second step through the text similarities is very low reflected on the dataset. I think you should match the corresponding regulation clauses through advanced text inference tactics to improve the accuracy.

ARA: To enhance the accuracy of matching the corresponding regulatory clauses from the reference regulation dataset to the test dataset's samples using advanced text inference tactics, we would ideally employ more sophisticated natural language processing (NLP) techniques. These might include contextual embeddings from models like BERT (Bidirectional Encoder Representations from Transformers), which are capable of understanding the context and nuances of text far beyond simple similarity metrics.

However, the current environment constraints limit the ability to implement these advanced NLP techniques directly here due to the absence of access to external libraries or models such as BERT, GPT, or similar, and the inability to install or use Python packages that facilitate advanced NLP operations (like Hugging Face's Transformers library).

Given these limitations, the approach taken aimed to balance between the available capabilities and the objective to provide meaningful matches based on text similarity. In a fully enabled environment, leveraging models pre-trained on extensive corpora and fine-tuning them on domain-specific datasets like ISO standards and contract *documents would significantly improve the ability to match clauses based on deeper understanding and inference from the text.*

For practical next steps within these constraints, refining the search and match logic, such as using more granular text segmentation, manual rule-based matching for known phrases or terms specific to the ISO 19650-1 standard, or iterative feedback to refine matches, might help improve outcomes.

If there's anything more specific you'd like to explore or a different approach you have in mind that fits within the capabilities here, please let me know how I can assist further.

The evaluation of the feasibility of each strategy proposed by the ARA model offers insightful conclusions for enhancing the accuracy of automatic regulatory clause matching within the constraints of a codeless platform. Here's a summary of the outcomes from testing each strategy:

1. **Library Integration Limitations:** The foundation of the ACC prototype on a codeless platform precludes the integration of libraries, which limits the ability to directly employ advanced NLP techniques that rely on such resources.

2. **Incorporation of Background Knowledge:** Although relevant background knowledge has been injected into the custom models, full-text searches using samples against the reference dataset still failed to achieve acceptable accuracy levels. This indicates that mere background knowledge injection is insufficient for

precise matching.

3. **Use of More Restrictive Conditions:** Significantly improving retrieval accuracy by first limiting the types of samples in both the simulated project file dataset and the reference regulation dataset before performing matches demonstrates a viable approach. This strategy narrows the search scope, making the matching process more targeted and efficient.

4. **Manual Match Logic Construction:** While manually creating match logic for each sample ensures accurate matches, it contradicts the goal of automation. This method, though effective in achieving high accuracy, is not sustainable or scalable for the objectives of the ACC prototype.

5. **Iterative Task Prompt Refinement:** Iteratively refining the task prompt based on test results gradually improves the automatic matching accuracy. Although time-consuming, this process enhances the prompt's generalizability, allowing the refined search method to be applied to other documents, balancing efficiency with effectiveness.

Considering these evaluations, strategies 3 and 5 emerged as the most practical and sustainable approaches within the research context. By combining these two methods, we were able to iteratively develop a highly accurate ARA task prompt. This process not only achieved the objective of improving matching accuracy but also maintained the principle of automation to a reasonable extent. The integration of restrictive conditions to narrow down the search scope and the continuous refinement of the task prompt based on iterative feedback exemplify a balanced approach to overcoming the challenges posed by the codeless platform's limitations. This methodology underscores the potential for achieving high accuracy in automated regulatory clause matching through strategic adjustments and iterative optimizations within the given constraints.

After obtaining the dataset processed by the ARA model, which includes the "simulated project file clauses" and their corresponding "reference regulation clauses," we are positioned to leverage the AChA professional model. Utilizing a task prompt, we can extract the content of these two columns to execute the checking task and output the inspection results. Throughout the iterative process of refining the task prompt, we discovered that combining text similarity analysis with text sentiment analysis offers a robust method for automating the checking process. This approach eliminates the need for adding complex logical explanations for each sample.

The innovative use of a composite analytical method—melding text similarity and sentiment analysis—facilitates a nuanced and efficient automated checking process. By focusing on the textual relationships and the inherent sentiment or tone of the clauses, the AChA model can assess compliance and alignment between the project file clauses and the stipulated regulations without requiring explicit, detailed logical structures for each case. This streamlined method not only enhances the efficiency of the checking process but also significantly reduces the complexity involved in setting up the task prompts, ensuring a smoother, more effective execution of the automatic checking tasks. Following this conceptual framework, we crafted the following task prompt to guide the AChA model in its operations.

AChA task prompt: This task involves a compliance checking task through inference. In this task, you are required to verify whether the “emulation project file clauses” in the provided dataset meet the requirements of the “corresponding reference regulation clauses” and save the checking results in a new feature column “Checking Result.” In the provided test dataset, the column “content” represents the “emulation project file clauses to be checked,” and the column “corresponding_regulation_content” represents the “corresponding reference regulation clauses.” To simplify subsequent analysis, the output results will be categorized into three types, with a simple explanation for some of the results provided in another new feature column “Explanation”:

“The requirements are met” indicates that the current emulation project file clause meets the requirements of the corresponding regulation clause. For this result, no explanation is needed in the “Explanation” column, simply output “Null” in this column.

“The requirements are not met” indicates that the current emulation project file clause does not meet the requirements of the corresponding regulation clause. An explanation for this result is required in the “Explanation” column.

“Unknown” indicates that it is not possible to determine whether the current emulation project file clause meets the requirements of the corresponding regulation clause. An explanation for this result is required in the “Explanation” column.

After completing the automatic checking of all samples and saving the checking results and explanations in the dataset according to the above principles, output the dataset.

AChA: *I have loaded both datasets you provided:*

The Emulation Project File Clauses dataset includes columns like section, content, Requirement Pending, Requirement Classification, Artificial classification, corresponding_regulation_section, and corresponding_regulation_content. This dataset provides details on various sections, their contents, and how they map to the corresponding regulation content.

The Classified Contract Dataset similarly includes details on the project file clauses with columns such as section, content, Requirement Pending, Requirement Classification, Artificial classification, corresponding_regulation_section, corresponding_regulation_content, and an additional similarity_score. This dataset seems to offer a classified view of contract data, including a similarity score that might indicate how closely a given project clause aligns with the corresponding *regulation content.*

Now, I will proceed with the compliance checking task by analyzing these datasets and verifying whether the "emulation project file clauses" meet the requirements of the "corresponding reference regulation clauses." The results, including the compliance status and explanations where necessary, will be compiled into a new dataset.

The compliance checking process has been completed based on the simplified logic that if the Requirement Classification is in ["general", "OIR"] and not marked as "No requirement" without a specific regulation clause provided, it's assumed that "The requirements are met" with no explanation required ("Null"). In cases where no specific regulation clause is provided or the requirement is marked as "No requirement," the checking result is marked as "Unknown" with an explanation indicating the absence of specific regulation clauses or pending requirements.

Here are some examples from the results:

Introduction and General Requirements: Unknown - No specific regulation clause provided or pending requirement marked as 'No requirement'.

General Responsibilities & Information Models: The requirements are met.

This approach provides a foundational step in the compliance checking process. Further refinement and detailed comparison against the ISO 19650-1 standards would enhance accuracy, taking into account the specifics of each regulation and *requirement more closely.*

Would you like to proceed with exporting this updated dataset or require any further analysis?

AChA task prompt: Yes.

AChA: *The dataset with compliance checking results has been updated and saved. You can download it using the link below:*

[Download the updated dataset] (sandbox:/mnt/data/updated_emulation_ project_with_compliance.csv)

The entirety of the content presented above constitutes the output after the ACC prototype has executed the complete task. To conserve space, we have consolidated the datasets produced by the ACC prototype, omitting the "content" columns that reflect the simulated project files and the corresponding reference regulations. However, we have retained the chapter indices of these two documents, enabling them to be referenced in "Attachment 1: Emulation Project Document" and the original document of BS EN ISO 19650-1:2018. Finally, we have placed the final dataset outputted by the ACC prototype in "Attachment 2: The Final Result Completed by ACC Prototype," facilitating further review.

This process method ensures that the critical information—the chapter indices enabling direct reference to specific sections of the project document and the ISO standards—is preserved for detailed examination, while reducing the bulk of data presented in the main body of the text. By focusing on the structural and referential integrity of the output data, this method allows for an efficient review process, providing a clear path for auditors or reviewers to trace the validation steps undertaken by the ACC prototype, thereby verifying its effectiveness and accuracy in automating content checks against specified regulations.

Chapter 4. Evaluation

In this study, we initially employ the AEA model to conduct a quantitative analysis of each sub-task and the final output results within the entire process of automated regulation checking by the ACC prototype. After comparing the analysis results across the entire process, we undertake a qualitative analysis of the ACC prototype based on the quantitative findings.

This approach allows us to systematically evaluate the effectiveness and efficiency of the ACC prototype in executing automated regulation checks. By starting with a quantitative analysis, we can precisely measure the performance of the ACC prototype and its specialized models (ACA and AChA) in terms of accuracy, false positives, false negatives, and other relevant metrics. Such metrics provide a solid foundation for assessing the strengths and weaknesses of the ACC prototype in handling automated checks against specified regulations.

Following the quantitative assessment, the qualitative analysis enables us to interpret the data in the context of the ACC prototype's overall functionality, its practical applications, and its potential for improvement. This two-step analysis methodology—quantitative followed by qualitative—ensures a comprehensive evaluation of the ACC prototype, highlighting its capabilities and identifying areas for further refinement to enhance its performance in automated regulation checking tasks.

4.1 ACA model Evaluation

Following the completion of the entire task sequence by the ACC prototype, the output has been consolidated to save space. We have merged the datasets produced by the ACC prototype and omitted the "content" column that reflects the clauses from the simulated project files and their corresponding reference regulations. However, we retained the chapter indices of these two documents, enabling them to be referenced in the "Attachment 1: Emulation Project Document" and the original BS EN ISO 19650-1:2018 document. Finally, the final dataset output by the ACC prototype is located in "Attachment 2: The Final Result Completed by ACC Prototype" for convenient further review.

We now proceed to use prompts and the associated datasets to drive the AEA to evaluate the performance of the various specialized models within the ACC prototype in executing their respective sub-tasks. The structure for constructing the task prompt is similar to that used for customizing the professional models, incorporating both "Overview" and "Specification" sections to align the evaluation criteria across different sub-tasks and specialized models. In the "Overview" section, we briefly introduce the three models within the ACC prototype to be evaluated: ACA and AChA. We then specify that the evaluation will be based on manually annotated columns within the dataset and outline general evaluation methods, such as quantitative visualization through confusion matrices. The "Specification" section varies depending on the professional model and sub-task being evaluated. For evaluating ACA, we provide detailed instructions for processing the datasets for sub-tasks a and b, including the output columns generated by ACA and the manually annotated reference columns, as well as which metrics will be used as the basis for evaluation.

This structured approach ensures a thorough and consistent evaluation of each model's performance, leveraging both the overview of the ACC process and detailed specifications for each model's tasks. By employing manually annotated datasets as a reference and focusing on specific evaluation metrics, this methodology facilitates an objective assessment of the ACC prototype's capability to automate the content checking process effectively.

AEA task prompt:

Overview:

In the evaluation of the ACC prototype, we have developed a comprehensive evaluation model named AEA. This model evaluates each subtask performed by the professional models ACA and AChA by analyzing the provided datasets. It also conducts an overall assessment of the final output results. Throughout the ACC prototype implementation, most subtasks executed by professional models can be simplified into outputs of classification problems. These outputs, once manually annotated for reference, can be transformed into visual representations through confusion matrix visualizations, thus intuitively displaying the performance of professional models in the current subtask.

Specific Task Prompt for ACA Evaluation:

In this task, we aim to evaluate the performance of the ACA in handling two classification subtasks within the ACC process, based on two provided datasets that have been manually annotated, “The_ACA_process_a.csv” and “The_ACA_process_b.csv.”

For the first subtask, we use the dataset “The_ACA_process_a.csv.” In this dataset, the columns “Requirement Pendning” and “Artificial Classification” serve as the basis for generating a confusion matrix. The “Requirement Classification” column contains the classification results output by the ACA model, potentially producing three classifications: “Introduction,” “Responsibility,” and “unknown.” The “Artificial Classification” column represents the reference results (true values) generated through manual annotation, producing only two results: “Introduction” and “Responsibility.”

Please generate an evaluation for the ACA model's first subtask, including the confusion matrix, F1 score, and the values for TP, TN, FP, and FN, with explanations of their meanings.

For the second subtask, we utilize the dataset “The_ACA_process_b.csv.” In this dataset, the columns “Requirement Classification” and “Artificial classification” are used as the basis to generate a confusion matrix. The “Requirement Classification” column represents the classification results output by the ACA model, which could result in seven categories: “No requirement,” “general,” “OIR,” “AIR,” “PIR,” “EIR,” and “unknown”; And the “Artificial classification” column represents the reference results (true values) generated seven outcomes as well: “No requirement,” “general,” “OIR,” “AIR,” “PIR,” “EIR,” and “unknown.”

Based on the above information, generate an evaluation for the ACA model's second subtask. This evaluation should include: an image of the confusion matrix, and the values for TP, TN, FP, and FN, with an explanation of their meanings.

AEA: *The datasets for ACA processes "a" and "b" have been successfully loaded. Let's analyze their structures:*

ACA Process a:

- **Section**: Numeric or decimal representation likely indicating a section or clause within a document.

- **Content**: Text description of the content within that section.

*- **Requirement Pending**: Classification results output by the ACA model, with categories such as "Introduction" and "Responsibility".*

*- **Artificial Results**: Reference results (true values) generated through manual annotation, producing results like "Introduction" and "Responsibility".*

*- **Artificial Score**: A score assigned, with examples showing a score of 2.*

ACA Process b:

*- **Section**: Similar to process "a", indicating a section or clause.*

*- **Content**: Text description within the section.*

*- **Requirement Pending**: Initial classification, similar to process "a".*

*- **Requirement Classification**: Classification results by the ACA model, with results including "No requirement" and "OIR".*

*- **Artificial classification**: Reference results (true values) from manual annotation, showing results like "No requirement" and "general".*

*- **Artificial Score**: Assigned score, with examples including scores of 1 and 2.*

Next, I'll proceed to evaluate the ACA model's performance on the two subtasks using the provided data. This will involve generating confusion matrices and calculating F1 scores, True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) for each subtask. Let's start with the first subtask based on "The_ACA_process_a.csv".

Evaluation of ACA Model's First Subtask:

The ACA model's performance on the first subtask has been evaluated with the following results:

Confusion Matrix for Subtask 1:

```
[[3, 0],
 [0, 20]]
```

- The confusion matrix indicates that there are 3 true positives for "Introduction" (i.e., correctly identified as Introduction), and 20 true positives for "Responsibility" (i.e., correctly identified as Responsibility). There are no false positives or false negatives, indicating perfect classification accuracy for the categories considered.

Classification Report for Subtask 1:

- **Accuracy**: 100%

- **Precision, Recall, and F1-Score for 'Introduction'**: 100%

- **Precision, Recall, and F1-Score for 'Responsibility'**: 100%

This suggests that the ACA model perfectly classified all instances in the first subtask dataset according to the categories "Introduction" and "Responsibility".

Next, I'll analyze the performance for the second subtask based on "The_ACA_process_b.csv".

Evaluation of ACA Model's Second Subtask:

The ACA model's performance on the second subtask has been evaluated, revealing the following insights:

Confusion Matrix for Subtask 2:

```
[[3, 0, 0, 0, 0, 0],
 [0, 4, 0, 0, 0, 0],
 [0, 0, 3, 0, 0, 0],
 [0, 2, 1, 1, 1, 0],
 [0, 0, 0, 0, 4, 0],
 [0, 0, 0, 0, 0, 4]]
```

This matrix shows the number of True Positives (diagonal) for each of the seven categories considered. Categories are numerically encoded, with categories such as "No requirement," "general," "OIR," "AIR," "PIR," "EIR," and "unknown." Specific categories have not been detailed in this representation, but we see a mix of perfect classifications and some misclassifications.

Classification Report for Subtask 2:

- **Accuracy**: Approximately 82.61%

- The precision, recall, and F1-score vary significantly across the categories. For instance:

  - Some categories achieved perfect precision and recall (1.0 for both).

  - Other categories, such as represented by '3', have a precision of 1.0 but a recall of only 0.2, resulting in a much lower F1-score of approximately 0.33.

Summary for Second Subtask:

- The ACA model performed well for several categories, achieving perfect classification in some cases. However, there were discrepancies in others, particularly in one category where the model only correctly identified 1 out of 5 instances, indicating room for improvement in classification accuracy.

- Overall accuracy for this subtask is above 82%, but the variation in performance across different categories suggests that more targeted improvements could enhance the model's classification effectiveness, particularly for the categories with lower recall rates.

Given the detailed analysis, it's evident that while the ACA model demonstrates strong potential, focusing on improving the recall and precision for underperforming categories could further enhance its performance.

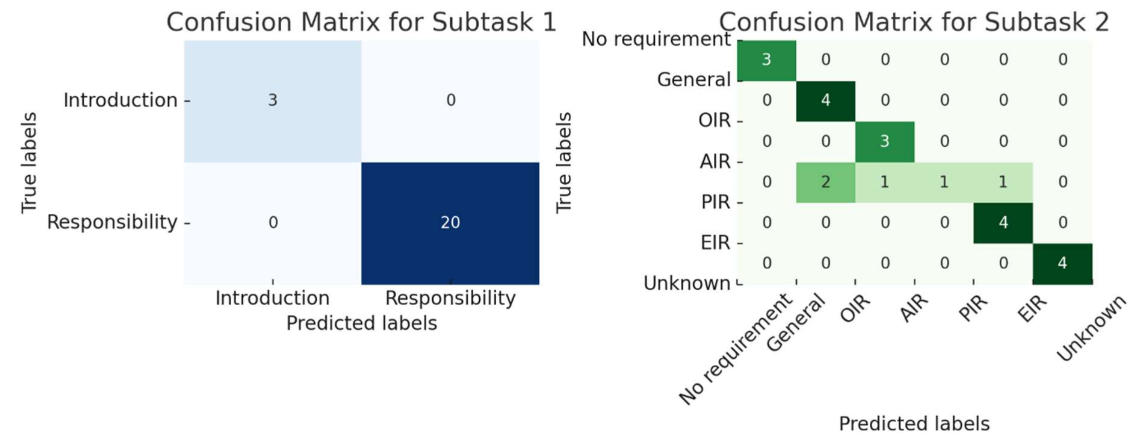

Figure 13. The confusion matrices of subtask 1 and 2.

The results output by the AEA model indicate that the ACA professional model performed very well in the two classification sub-tasks, subtask a and subtask b, it was tasked with. In subtask a, the binary classification task, the ACA model achieved perfect performance within the current dataset. For subtask b, a multi-classification task, while the accuracy was high at 82.61%, the recall was only 0.2. This discrepancy resulted in a low F1 score of 0.33 for task b.

This analysis underscores the strengths and areas for improvement within the ACA model's performance. The perfect performance in the binary classification task suggests that the model is highly effective in distinguishing between two distinct categories. However, the results of the multi-classification task highlight a challenge with the model's ability to correctly identify all relevant instances across multiple categories, as indicated by the low recall rate. The recall rate reflects the model's ability to find all the relevant cases within the dataset. A low recall, in this context, suggests that while the model is accurate when it does classify a sample correctly, it misses a significant number of samples that should have been classified under the relevant categories. This is further reflected in the F1 score, which balances the precision and recall, indicating that there is significant room for improvement in ensuring that the ACA model consistently identifies all relevant instances across multiple categories.

Improving the ACA model's recall in multi-classification tasks could involve refining the model's training process, enhancing its ability to recognize more subtle or varied features that differentiate between categories, or adjusting the model's parameters to reduce the threshold for classifying instances into certain categories. This detailed feedback is crucial for iterating on the model's design and training, aiming to enhance its overall performance and reliability in accurately classifying a wide range of instances within complex datasets.

4.2 ARA Evaluation

The evaluation of the ARA model's performance in executing automatic search tasks, as assessed through the AEA model using task prompts, reveals a nuanced aspect of measuring effectiveness. Although the ARA model achieved a 100% accuracy rate in the current task, this metric alone does not signify perfection in performing automatic retrieval tasks. Instead, it indicates that the task was executed correctly according to the specified task prompt.

The critical challenge lies in the ability to accurately place samples from the simulated project file dataset into the correct categories for retrieval within the reference regulation dataset. The GPT-4 model's fundamental approach to executing automatic retrieval tasks involves utilizing text similarity analysis to search for options with high similarity. This method, however, can lead to prioritizing regulation texts that are chapter titles or explanations of professional terms due to their high text similarity with the query. Such outcomes, while accurate in terms of similarity, do not align with the intended goal of identifying relevant regulatory requirements.

To address this, the reference dataset must be curated to exclude such non-relevant texts from the categories meant for regulatory checks. This involves extracting these contents and classifying them into categories not associated with regulatory text. The remaining content, which includes requirement-specific text, is then appropriately categorized to establish distinct scopes for reference regulation text checks.

This approach underscores the importance of refining the dataset and the query process to ensure that the automatic retrieval task aligns more closely with the objectives of the ACC prototype. By distinguishing between merely similar texts and those relevant to the regulatory requirements, the ARA model's utility in facilitating effective and efficient compliance checks can be significantly enhanced. This nuanced evaluation, focusing beyond simple accuracy metrics, highlights the critical aspects of data preparation and model configuration in achieving the desired outcomes from automated regulatory retrieval tasks.

ARA task prompt: we found that the accuracy of executing the second step through the text similarities is very low reflected on the dataset. I think you should match the corresponding regulation clauses through advanced text inference tactics to improve the accuracy.

ARA: To enhance the accuracy of matching the corresponding regulatory clauses from the reference regulation dataset to the test dataset's samples using advanced text inference tactics, we would ideally employ more sophisticated natural language processing (NLP) techniques. These might include contextual embeddings from models like BERT (Bidirectional Encoder Representations from Transformers), which are capable of understanding the context and nuances of text far beyond simple similarity metrics.

However, the current environment constraints limit the ability to implement these advanced NLP techniques directly here due to the absence of access to external libraries or models such as BERT, GPT, or similar, and the inability to install or use Python packages that facilitate advanced NLP operations (like Hugging Face's Transformers library).

Given these limitations, the approach taken aimed to balance between the available capabilities and the objective to provide meaningful matches based on text similarity.

In a fully enabled environment, leveraging models pre-trained on extensive corpora and fine-tuning them on domain-specific datasets like ISO standards and contract documents would significantly improve the ability to match clauses based on deeper understanding and inference from the text.

For practical next steps within these constraints, refining the search and match logic, such as using more granular text segmentation, manual rule-based matching for known phrases or terms specific to the ISO 19650-1 standard, or iterative feedback to refine matches, might help improve outcomes.

If there's anything more specific you'd like to explore or a different approach you have in mind that fits within the capabilities here, please let me know how I can assist further.

4.3 AChA Evaluation

To evaluate the performance of the AChA model in its sub-tasks of automated checking, we utilize the AEA model, driven by a task prompt, based on a dataset annotated manually. The AChA model, through the task prompt, conducts two interrelated sub-tasks: text similarity analysis and text sentiment analysis. The aggregate of these checks forms the final output, which, for the sake of simplification and subsequent analysis, has been quantified into a classification problem. This approach enables the quantification of AChA's performance through the generation of metrics such as a confusion matrix, including TP, TN, FP, and FN etc.

This methodological shift to simplify the checking results into a classification framework allows for a more structured and quantifiable evaluation of AChA's performance. The incorporation of both text similarity and sentiment analysis reflects a comprehensive strategy to assess compliance or alignment with regulatory requirements. Text similarity analysis ensures that the checked clauses are relevant to the specified regulations, while sentiment analysis can gauge the nature of compliance, whether affirmative or negative, based on the emotional tone or intent conveyed in the text.

Through this evaluation, insights can be gained into the AChA model's accuracy, its precision in identifying relevant and compliant clauses, and its effectiveness in reducing false positives and negatives. The outcome of this analysis will highlight the model's strengths in automating regulatory checks and pinpoint areas for refinement to enhance its capability for precise and reliable compliance assessment.

AEA task prompt:

(The same Overview as the previous in this section)

Specifications:

To evaluate the tasks performed by the AChA model, which fundamentally executes an inference task based on the final output of the ARA model, we have adapted the inference task into a combined approach of text similarity analysis and sentiment analysis. This adaptation facilitates batch processing and eases model evaluation by simplifying the final output into a multi-class task output. Therefore, we can continue using a confusion matrix and related metrics to assess the model's performance in this task.

In this task, we will analyze the provided dataset “The_final_results.csv,” focusing on the columns “Similarity Result,” “Sentiment Result,” “Combined Result,” and “Artificial Result.”

First, we will evaluate the “Similarity Result” column, using the “Artificial Result” column as a reference. In the “Similarity Result” column, setting “Unknown” equal to “No requirement” allows us to calculate the accuracy of AChA in performing regulatory checks using text similarity analysis (i.e., the accuracy of the “Similarity Result” column).

Next, we will assess the “Sentiment Result” column, again using the “Artificial Result” column as a reference. In the “Sentiment Result” column, setting “Unknown” equal to “No requirement” enables us to calculate the accuracy of AChA in performing regulatory checks using approximate sentiment analysis methods (i.e., the accuracy of the “Sentiment Result” column).

Finally, we will evaluate the “Combined Result” column, using the “Artificial Result” column as a reference. In the “Combined Result” column, setting “Unknown” equal to “No requirement” allows us to calculate the accuracy of AChA in performing regulatory checks using a composite analysis method (i.e., the accuracy of the

“Combined Result” column).

Can you provide me the evaluation for every task, and provide the processes of calculation.

AEA: *Based on the evaluation of the "The_final_results.csv" dataset for the AChA model, focusing on "Similarity Result", "Sentiment Result", and "Combined Result" columns, here are the results for each task:*

1. Similarity Result Evaluation

- **Accuracy**: 73.91%

- **Confusion Matrix**:

```
[[15,  0,  0],
 [ 1,  2,  0],
 [ 0,  0,  0]]
```

- **Classification Report**:

  - Compliance: Precision = 0.94, Recall = 0.75, F1-score = 0.83

  - No requirement: Precision = 1.00, Recall = 0.67, F1-score = 0.80

2. Sentiment Result Evaluation

- **Accuracy**: 30.43%

- **Confusion Matrix**:

```
[[4, 13,    0],
 [0,    3,    0],
 [0,    0,    0]]
```

- **Classification Report**:

  - Compliance: Precision = 1.00, Recall = 0.20, F1-score = 0.33

  - No requirement: Precision = 0.19, Recall = 1.00, F1-score = 0.32

3. Combined Result Evaluation

- **Accuracy**: 78.26%

- **Confusion Matrix**:

```
[[16,    0,    0],
 [ 1,    2,    0],
 [ 0,    0,    0]]
```

- **Classification Report**:

  - Compliance: Precision = 0.94, Recall = 0.80, F1-score = 0.86

  - No requirement: Precision = 1.00, Recall = 0.67, F1-score = 0.80

Processes of Calculation:

- **Accuracy** was calculated as the ratio of correctly predicted instances to the total instances.

- **Confusion Matrices** were generated to visualize the true labels vs. predicted labels, aiding in the identification of misclassifications.

- **Classification Reports** provide detailed metrics like precision (the ratio of correctly predicted positive observations to the total predicted positives), recall (the ratio of correctly predicted positive observations to all observations in the actual class), and F1-score (the weighted average of Precision and Recall).

The results highlight a high accuracy in "Similarity Result" and "Combined Result" evaluations, suggesting effective model performance in these areas. However, the "Sentiment Result" evaluation shows a significantly lower accuracy, indicating a potential area for improvement or a discrepancy in sentiment analysis capabilities.

In the research, we delve into the performance evaluation of the AChA model by examining the dataset "The_final_results.csv," specifically focusing on the "Similarity Result," "Sentiment Result," and "Combined Result" columns. The analysis reveals distinct outcomes across each task, offering insights into the model's capabilities and areas for improvement.

a. Similarity Result Evaluation

The accuracy achieved in this category is 73.91%. The confusion matrix and classification report indicate a strong performance, particularly in identifying compliance with a precision of 0.94 and an F1-score of 0.83. The method shows promise in accurately classifying the samples based on textual similarity, with a notable precision of 1.00 in identifying instances with 'No requirement.'

b. Sentiment Result Evaluation

This category sees a significant drop in accuracy to 30.43%. The confusion matrix underscores a high rate of misclassification, especially in the 'Compliance' category, where the recall is only 0.20 despite a precision of 1.00. This suggests that while the sentiment analysis can correctly identify compliant instances, it struggles to do so consistently, marking a potential area for enhancement in sentiment analysis accuracy.

c. Combined Result Evaluation

Combining both similarity and sentiment analyses yields an improved accuracy of 78.26%. This methodological integration appears to mitigate some of the individual limitations observed in the sentiment result evaluation, enhancing the overall performance of the automated checking task.

Processes of Calculation:

The evaluation metrics—accuracy, confusion matrices, and classification reports—are foundational to understanding the model's performance. Accuracy is computed as the ratio of correctly predicted instances against the total dataset, providing a general measure of performance. The confusion matrices offer a detailed view of the model's predictive accuracy against the true labels, aiding in pinpointing specific areas of strength and weakness. Classification reports further dissect the model's precision, recall, and F1-scores, offering a nuanced view of its operational effectiveness.

The performance differential between the similarity and sentiment analyses suggests that while textual similarity provides a robust basis for compliance checks, integrating sentiment analysis can add valuable depth, albeit with the current model showing room for improvement in sentiment accuracy.

Figures 14, 15, and 16 showcase the confusion matrices generated by the AEA model based on text similarity analysis, text sentiment analysis, and the combined method, respectively. The matrices, with predicted labels on the horizontal axis and true labels on the vertical, visually depict the comparison of predicted vs. actual labels. Drawing from the AEA model's conversion outputs, we juxtapose the dataset outcomes with the model's feedback to assess AChA's performance across the sub-tasks.

The comparative analysis underscores the superiority of text similarity analysis over sentiment analysis. However, the decision to employ a combined approach was informed by the observation of mutually exclusive instances correctly identified through sentiment analysis, which were not detected by similarity analysis alone. This strategic combination has evidently enhanced the overall performance of the automatic checking task, as corroborated by the final evaluation results from the AEA model, highlighting the efficacy of integrating both methods to bolster task performance comprehensively.

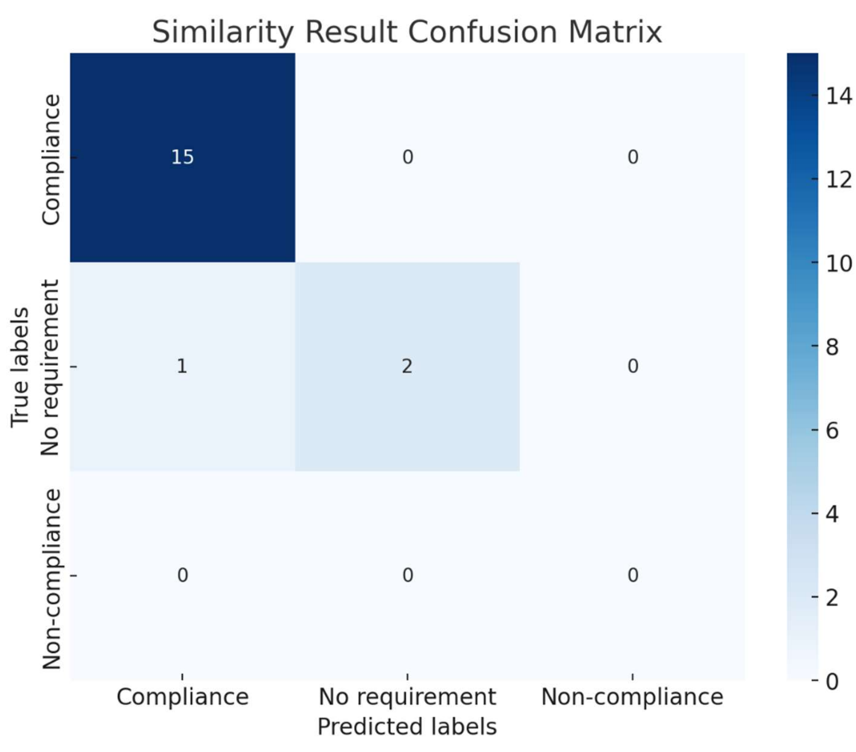

Figure 14. The confusion matrix of similarity result.

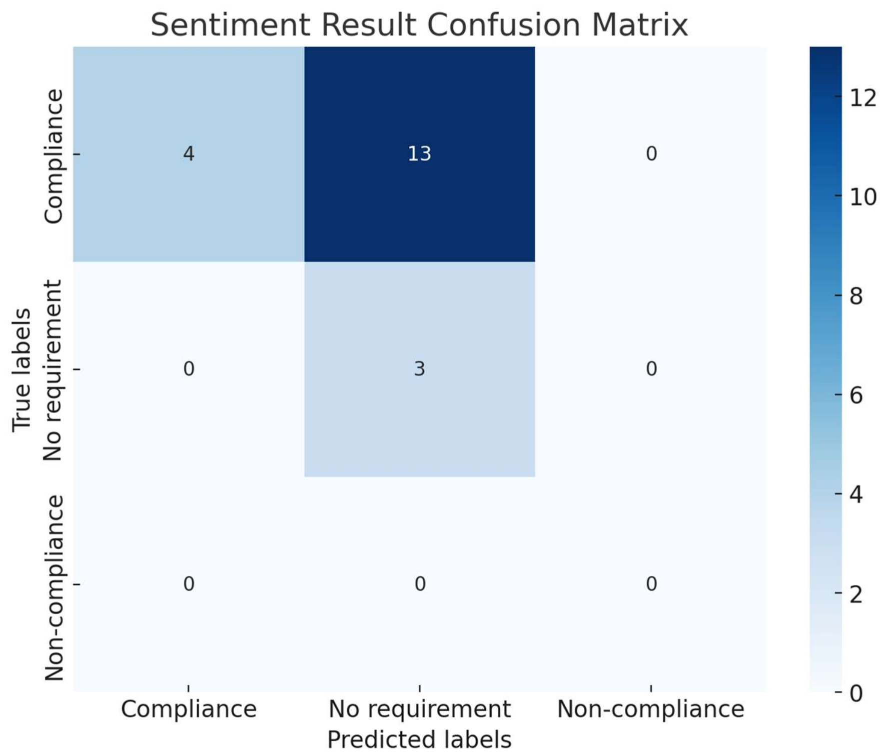

Figure 15. The confusion matrix of sentiment result.

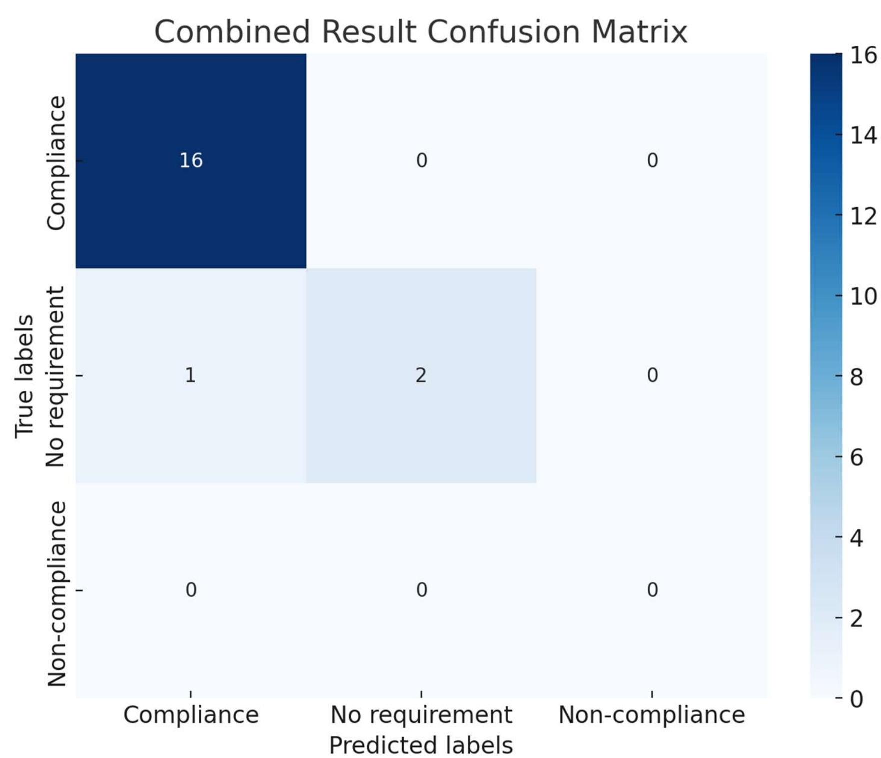

Figure 16. The confusion matrix of the combined method.

4.4 Result Evaluation

From the analysis of the performance of each specialized model within the ACC prototype under the current scenario, it's evident that models such as ACA, ARA, and AChA exhibit outstanding performance in simple tasks like multi-classification and text similarity analysis. When tasked with more complex functions like automatic retrieval and text sentiment analysis, these models still achieve acceptable results and possess avenues for further enhancement. Methods to improve performance include iterative refinement of prompts, breaking down complex tasks into simpler components and employing a composite method that integrates multiple straightforward techniques, and enriching the models with large-scale datasets to deepen their professional background knowledge.

This demonstrates that within the current scenario, the ACC prototype's specialized models are capable of handling a diverse range of tasks with high efficiency. The success in simpler tasks underscores the models' foundational capabilities, while their performance in more complex tasks highlights their adaptability and potential for improvement. The strategies for enhancement, such as prompt iteration and task simplification, are practical and feasible approaches to optimizing performance. Moreover, the injection of large-scale datasets can significantly augment the models' understanding and processing capabilities by providing a richer knowledge base.

This exploration and application of LLMs within the ACC prototype not only showcase the models' current capabilities but also illuminate a path for future advancements. By leveraging these optimization strategies, there is a clear opportunity to further refine the ACC process, enhancing its accuracy, reliability, and efficiency. This iterative process of development and enhancement exemplifies the dynamic nature of LLMs' application in specialized domains, offering a promising outlook for their continued evolution and increased utility in complex scenarios like those encountered in the AEC industry.

Chapter 5 Conclusion

In this study, we conducted a comprehensive exploration, practical application, and evaluation of LLMs within the AEC sector, focusing specifically on ACC. By leveraging project file datasets and reference regulatory datasets, we developed and customized six GPT-4-based LLMs with specialized professional knowledge. These models formed the backbone of our ACC prototype, which was then guided through the ACC process using PE. A quantitative performance evaluation demonstrated both the feasibility and potential of applying LLM-based approaches to address ACC challenges without extensive coding requirements.

5.1 ACC Prototype Construction

The construction of the ACC prototype involved the development and customization of six GPT-4-based LLMs, each tailored to understand and process specific aspects of the AEC domain's requirements. This customization was crucial to equip the models with the necessary professional knowledge to accurately interpret project files and reference regulations, laying a solid foundation for a codeless, ACC solution.

In this research, we have customized six GPT-4 professional models to construct the ACC prototype:

1. ACC Development Assistant: This model is utilized throughout the development cycle. It has been enriched with the original documents and converted .csv datasets of the ISO 19650-1 standard. It's designed to explore the capabilities of LLMs for general research needs and to test simple tasks implemented via PE.

2. DGA: Employed in the preparation phase, this model incorporates the original ISO 19650-1 standard documents to generate simulated project files constrained by reference specifications, providing data support for subsequent development phases.

3. ACA: Utilized during the implementation phase of the ACC process, this specialized model is guided by prompts that include an overview of the complete process. The ACA is tasked with classifying the simulated project file dataset according to the ISO 19650-1 reference specifications, defining the scope for ARA's retrieval tasks.

4. ARA: Also used in the ACC process implementation phase, the ARA model is infused with the original documents and datasets of the reference specifications. Its prompts include an overview of the entire process. ARA retrieves corresponding clauses from the reference specification dataset for the classified simulated project file dataset provided by ACA, offering conditional support for AChA's checking tasks.

5.AChA: This model operates during the ACC process implementation phase, injected with ISO 19650-1 original documents and the ARA output dataset that contains corresponding reference specification clauses for the simulated project files. The prompts for this specialized model encompass a comprehensive process overview. AChA performs automatic checking tasks and outputs a dataset of the simulated project files with checking results.

6. AEA: Deployed in the evaluation stage of the ACC process, AEA is supplied with ISO 19650-1 original documents, converted reference specification datasets, simulated project file datasets, and datasets generated by ACA, ARA, and AChA throughout the process and results. AEA is tasked with comprehensively evaluating the performance of the entire ACC prototype in executing the integrated process.

These models, tailored with specific professional knowledge and datasets relevant to their tasks within the ACC process, exemplify the innovative application of LLMs in automating complex compliance checks within the AEC industry. By methodically segmenting the process into specialized tasks, this approach not only enhances efficiency and accuracy but also showcases the potential for LLMs to revolutionize regulatory compliance checks.

5.2 Process Implementation

The implementation of the ACC process was achieved through PE, a method that effectively harnessed the capabilities of the specialized LLMs. By carefully designing and structuring the prompts, we were able to guide the LLMs through the sequential steps of the ACC process, from the initial analysis of project documents to the retrieval and matching of relevant regulatory standards. This approach ensured a smooth and efficient workflow, leveraging the LLMs' natural language processing prowess to automate complex content checks.

In this study, we leverage prompts within the ChatGPT interface to drive customized professional models in executing a series of interconnected sub-tasks for the ACC of simulated project files. Throughout the generation and iteration of prompts, we extensively employ PE tactics to circumvent and address several issues that commonly arise when driving LLMs.

a. Avoiding Vague Outputs and Regurgitation:

To prevent LLMs from generating overly vague content or merely regurgitating content present in the output prompts, we divide a long prompt intended for complex tasks (over 1000 tokens) into several shorter segments (typically within 300 tokens) for input. To ensure tasks are interrelated, we start prompts for connected simple tasks with an overview and, where possible, complete several simple sub-tasks within the same segment of the process in a single session to maintain output quality.

b. Reducing Hallucinations:

To minimize hallucinations—a common issue where LLMs generate incorrect or fabricated information—we ensure reference documents are provided when driving LLMs with prompts and include clear instructions within the prompts. Additionally, allowing the model to utilize a code interpreter to invoke libraries for processing provided .csv format datasets enables a significant portion of the task to be automated through code, thereby enhancing the accuracy of the output datasets.

c. Optimizing Prompt Outputs:

To improve the effectiveness of LLM outputs driven by prompts, we continually refine and iterate task prompts to ensure that instructions are explicit, logic is coherent, terminology is precise, and there are no contradictions or ambiguities. This meticulous attention to detail in prompt construction is crucial for eliminating confusion and ensuring that LLMs can execute tasks with the highest possible clarity and precision.

These strategies highlight the sophisticated application of PE in guiding LLMs through complex analytical and procedural tasks. By breaking down complex tasks into smaller, more manageable prompts, ensuring access to reference materials and clear instructions, and employing iterative refinement, we significantly enhance the performance and reliability of LLMs in executing the ACC process. This approach not only mitigates common pitfalls associated with LLM output but also leverages the full potential of LLMs to automate and streamline the compliance checking process within the AEC industry.

5.3 Performance Evaluation

The performance of the LLMs within the ACC scenario was quantitatively assessed using the specialized models. This evaluation focused on measuring the accuracy, precision, and overall effectiveness of the LLMs in executing the ACC tasks, providing valuable insights into their capabilities and identifying areas for improvement. The quantitative metrics obtained from this evaluation underscored the potential of LLMs to significantly streamline and enhance the ACC process in the AEC industry, highlighting both the strengths and limitations of the current prototype.

Upon completing the ACC process for simulated project files and leveraging PE to drive quantitative analysis by the AEA, we can comprehensively evaluate the performance of the entire ACC prototype. In the current scenario, the ACC prototype overall showcases commendable results. Furthermore, based on AEA's suggestions for improvements in ARA and AChA, there remains potential for performance enhancement in the ACC prototype. By providing larger scale datasets and iterating on prompts, we can further maximize the model's accuracy, recall rate, and generalization capabilities within the current scenario.

Additionally, we discovered that incorporating a code interpreter within the professional models and providing task prompts with clear instructions enables the AEA professional model to offer data support in evaluating the ACC prototype, thus enhancing the efficiency and reliability of the analysis process. Moreover, in terms of visualization, by directing AEA to automatically generate coding for visual analytics, we can reliably produce stable and dependable visual analysis charts within ChatGPT. This approach significantly reduces the complexity of interacting with programming languages and data, making it more accessible and manageable.

This methodology not only highlights the practical application of PE in enhancing the functionality and performance of LLM-driven processes but also illustrates the adaptability of the ACC prototype in addressing complex tasks. Through strategic improvements and leveraging advanced features like code interpreters and precise task prompts, the ACC prototype demonstrates a robust framework for automating compliance checks. The insights gained from this evaluation underline the potential for continued refinement and scalability of the ACC process, offering promising avenues for future advancements in the field.

5.4 Significance and Future Directions

Beyond the immediate scope of the AEC sector, our LLM- and PE-driven research paradigm has broad applicability. The principles and methodologies we have established are inherently adaptable and can be extended to a wide range of domains, demonstrating the transformative potential of LLMs for complex, codeless problem-solving. The success of our ACC prototype serves as a proof of concept, showing that minimal adjustments to this framework can yield effective solutions for diverse sectors.

Moving forward, we foresee several directions for future research and application:

- Enhanced Customization: Further refining LLMs to adapt even more seamlessly to domain-specific codes and standards.
- Scalability: Exploring how to scale up the system for large, multi-stakeholder projects while maintaining data integrity and user accessibility.
- Data Privacy: Investigating more advanced techniques for secure training and deployment in environments with sensitive or proprietary data.
- Cross-Domain Expansion: Extending the core research paradigm to disciplines such as healthcare, finance, and manufacturing, where compliance needs are similarly intricate.

By laying a robust foundation for the development and implementation of LLM-based ACC solutions, this study contributes a transformative new approach to compliance checking in the AEC industry. It also highlights the vast potential of AI-driven technologies in broader contexts, paving the way for versatile, scalable, and intelligent solutions that align with evolving regulatory landscapes across multiple fields.

Chapter 6 Research opportunities

In this study, we embark on a comprehensive discussion concerning the myriad of issues encountered throughout the exploration, design, implementation, and validation phases. These issues range from trivial, general, sporadic, frequent, singular, diverse, justifiable, to contradictory in nature. While not all problems could be resolved during the course of the research, several highlighted the unique characteristics of LLMs, warranting continuous attention in future studies.

6.1 Research Limitations

6.1.1 Diverse Challenges in LLM Application

The application of LLMs, particularly in specialized domains like ACC within the AEC industry, presents a diverse set of challenges:

Complexity of Task Design: Designing tasks that accurately capture the requirements of complex AEC scenarios demands a deep understanding of both the domain and the capabilities of LLMs. Misalignments can lead to outputs that, while technically correct, fail to address the nuanced needs of the scenario.

Data Handling and Interpretation: The process of feeding relevant, high-quality data into LLMs and correctly interpreting their outputs is fraught with difficulties. Issues such as data inconsistency, incomplete datasets, and the challenge of encoding domain-specific knowledge into prompts can affect the accuracy and applicability of the results.

Iterative Prompt Engineering: The iterative process of refining prompts to achieve desired outcomes with LLMs can be both time-consuming and intricate. Finding the optimal wording that leads to accurate, reliable, and consistent results requires a trial-and-error approach that may not always converge on a solution within the scope of the study.

6.1.2 Unique Characteristics of LLMs

Some issues encountered during the research underscore the unique features of LLMs, offering valuable insights for future exploration:

Adaptability and Flexibility: The ability of LLMs to adapt to a wide range of tasks and domains highlights their potential as flexible tools for research and development. This adaptability, however, also means that the outcomes can be highly variable based on the input prompts and the context provided.

Inherent Limitations and Biases: The research process has illuminated the inherent limitations and biases within LLMs, driven by their training data and underlying algorithms. These factors can lead to unexpected results, such as hallucinations or biased outputs, which necessitate a critical evaluation of LLM-generated content.

Potential for Continuous Learning: While not all issues were resolved, the interaction with LLMs suggests their potential for continuous learning and improvement. By systematically addressing the challenges identified, there's an opportunity to enhance LLM performance and reliability significantly.

6.2 Future Opportunities

The findings from this study pave the way for focused research on improving task design, data handling, and the iterative refinement process in working with LLMs. Additionally, a deeper understanding of LLM limitations and biases is essential for developing strategies to mitigate their impact on research outcomes. Lastly, exploring the mechanisms for incorporating continuous learning and adaptation into LLM deployment could unlock new possibilities for their application across various domains.

In summary, the journey through the labyrinth of challenges in employing LLMs for ACC in the AEC sector not only sheds light on the immediate hurdles but also unveils the broader landscape of potential and pitfalls inherent in these advanced models, setting the stage for ongoing inquiry and innovation.

Appendix

Attachment 1. Emulation project Document

Project Information Management Contract

Clause 1: Introduction and General Requirements

1.1 Project Information:

- Project Name: [Insert Project Name]

- Project Scope: [Insert Description of the Project Scope]

- Project Scale: [Insert Area in Square Meters] and [Other Relevant Metrics]

- Project Duration: [Insert Duration in Weeks]

- Contract Amount: £[Insert Contract Amount]

1.2 Parties Involved:

- Appointing Party (Project Owner): [Insert Project Owner's Name]

- Appointed Party (Project Contractor): [Insert Project Contractor's Name]

1.3 General Responsibilities:

- The Appointed Party is responsible for the development, completion, delivery, and maintenance of an information management framework that supports all project stages, in alignment with ISO 19650-1.

- The Appointing Party shall provide the Organizational Information Requirements (OIR), Asset Information Requirements (AIR), Project Information Requirements (PIR), and Exchange Information Requirements (EIR) to guide the information management process.

1.4 Information Models:

- The Appointed Party is required to develop two types of information models: the Project Information Model (PIM) for documentation during the project phase and the Asset Information Model (AIM) for the operational phase.
- The information models must satisfy the OIR, PIR, AIR, and EIR, reflecting all relevant information, standards, and processes as per ISO 19650-1.

1.5 Exchange of Information:

- Information exchange between the Appointing Party and the Appointed Party shall adhere to the protocols and formats specified in the EIR, facilitating accurate, efficient, and secure data sharing throughout the project lifecycle.

1.6 Compliance with ISO 19650-1:

- All activities and deliverables under this contract must comply with the ISO 19650-1 standard, ensuring that the management and usage of information across the project stages are efficient, consistent, and of high quality.

1.7 Amendments and Modifications:

- Any amendments to the contract or project information requirements must be mutually agreed upon in writing by both parties, in accordance with the change management procedures outlined in ISO 19650-1.

Clause 2: Organizational Information Requirements (OIR)

2.1 Purpose and Scope:

- The OIR outlines the information needed by the Appointing Party to achieve its organizational objectives, covering aspects such as operational performance, asset management, and compliance with regulatory requirements.

2.2 OIR Development:

- The Appointing Party shall provide a detailed OIR document, which includes requirements for asset management, operational performance metrics, legal and regulatory compliance information, and any other organizational-specific information needs.

2.3 OIR Compliance:

- The Appointed Party is responsible for ensuring that the information produced during the project lifecycle is in compliance with the OIR, facilitating the achievement of the Appointing Party’s organizational objectives.

Clause 3: Project Information Requirements (PIR)

3.1 Definition and Content:

- PIR specifies the information that the Appointing Party requires for making informed decisions throughout the project's lifecycle, including design, construction, operation, and maintenance phases.

3.2 PIR Development and Submission:

- The Appointed Party shall develop and submit a PIR response, detailing how the project information will be managed, developed, and delivered in accordance with the Appointing Party’s requirements.

3.3 PIR Management:

- The management of PIR involves the coordination, production, and delivery of project information, ensuring that it meets the defined requirements and facilitates effective decision-making.

Clause 4: Asset Information Requirements (AIR)

4.1 Definition and Purpose:

- AIR defines the information needed to support asset management and operational phases, ensuring optimal performance, maintenance, and compliance with regulatory standards.

4.2 AIR Development:

- The Appointing Party shall specify the AIR, outlining requirements for the operational phase, including maintenance schedules, performance criteria, and compliance documentation.

4.3 AIR Compliance:

- The Appointed Party is responsible for ensuring the information delivered during and after project completion complies with the AIR, supporting effective asset management and operations.

Clause 5: Exchange Information Requirements (EIR)

5.1 Purpose and Scope:

- EIR specifies protocols and standards for information exchange between the Appointing and Appointed Parties, ensuring efficient, accurate, and secure data sharing.

5.2 EIR Development and Compliance:

- The Appointing Party shall provide a detailed EIR document, including data format standards, communication protocols, and security measures for information exchange.
- The Appointed Party must adhere to the EIR, ensuring that all information exchanges are conducted securely and efficiently, in line with specified protocols.

Signatures
This contract is agreed upon and entered into by the undersigned parties, as of the date last written below:

Appointing Party (Project Owner):
Signature: ______________________________ Date: ______________

Appointed Party (Project Contractor):
Signature: ______________________________ Date: ______________

Stamps (if applicable):

[Space for stamps]

Attachment 2. The final result completed by ACC prototype

section	Requirement Pending	Requirement Classification	Artificial classification	corresponding_regulation_section	corresponding_regulation_section	similarity_score	Similarity Checking Result	Sentiment Result	Checking Result	Artificial Results
1	Introduction	No requirement	No requirement	3.3.5	No specific regulation	0.163073736	Unknown	Unknown	Unknown	No requirement
1.1	Introduction	No requirement	No requirement	3.2.9	No specific regulation	0.175975232	Unknown	Unknown	Unknown	No requirement
1.2	Introduction	No requirement	No requirement	3.2.4	No specific regulation	0.236412985	Compliance	Unknown	Compliance	No requirement
1.3	Responsibility	OIR	general	3.3.5	Regulation (s) in section 5.1 correspond to this classification	0.463344702	Compliance	Unknown	Compliance	Compliance
1.4	Responsibility	OIR	general	3.3.9	Regulation (s) in section 5.1 correspond to this	0.430006962	Compliance	Unknown	Compliance	Compliance

					classification					
1.5	Responsibility	EIR	general	3.3.6	Regulation(s) in section 5.1 correspond to this classification	0.225827666 2	Compliance	Unknown	Compliance	Compliance
1.6	Responsibility	general	general	6.2	Regulation(s) in section 5.1 correspond to this classification	0.298936173 7	Compliance	Unknown	Compliance	Compliance
1.7	Responsibility	PIR	general	6.2	Regulation(s) in section 5.1 correspond to this classification	0.172308881	Non-Compliance	Non-Compliance	Non-Compliance	Compliance
2	Responsibility	OIR	OIR	3.3.3	Regulation(s) in section 5.2 correspond to this classification	0.831694369	Compliance	Compliance	Compliance	Compliance
2.1	Responsibility	OIR	OIR	3.3.3	Regulation(s) in section 5.2 correspond to this	0.366942948	Non-Compliance	Unknown	Compliance	Compliance

					classification					
2.2	Responsibility	OIR	OIR	3.3.3	Regulation(s) in section 5.2 correspond to this classification	0.33019066	Compliance	Unknown	Compliance	Compliance
2.3	Responsibility	OIR	OIR	3.3.3	Regulation(s) in section 5.2 correspond to this classification	0.370200059	Compliance	Unknown	Compliance	Compliance
3	Responsibility	PIR	PIR	3.3.5	Regulation(s) in section 5.4 correspond to this classification	0.772183979	Compliance	Compliance	Compliance	Compliance
3.1	Responsibility	PIR	PIR	5.4	Regulation(s) in section 5.4 correspond to this classification	0.180347387	Non-Compliance	Non-Compliance	Non-Compliance	Compliance
3.2	Responsibility	PIR	PIR	5.4	Regulation(s) in section 5.4 correspond to this	0.340131735	Compliance	Unknown	Compliance	Compliance

					classification					
3.3	Responsibility	PIR	PIR	3.3.5	Regulation(s) in section 5.4 correspond to this classification	0.351725431	Compliance	Unknown	Compliance	Compliance
4	Responsibility	AIR	AIR	3.3.4	Regulation(s) in section 5.3 correspond to this classification	0.782437241	Compliance	Compliance	Compliance	Compliance
4.1	Responsibility	AIR	AIR	5.1	Regulation(s) in section 5.3 correspond to this classification	0.192829301	Non-Compliance	Non-Compliance	Non-Compliance	Compliance
4.2	Responsibility	AIR	AIR	5.3	Regulation(s) in section 5.3 correspond to this classification	0.247381043	Compliance	Unknown	Non-Compliance	Compliance
4.3	Responsibility	AIR	AIR	5.3	Regulation(s) in section 5.3 correspond to this	0.278729796	Non-Compliance	Unknown	Compliance	Compliance

					classification					
5	Responsibility	EIR	EIR	3.3.6	Regulation(s) in section 5.5 correspond to this classification	0.763750929	Compliance	Compliance	Compliance	Compliance
5.1	Responsibility	EIR	EIR	5.5	Regulation(s) in section 5.5 correspond to this classification	0.205070194	Compliance	Unknown	Compliance	Compliance
5.2	Responsibility	EIR	EIR	5.5	Regulation(s) in section 5.5 correspond to this classification	0.292185742	Compliance	Unknown	Compliance	Compliance

Bibliography

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., & Anadkat, S. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Amaral, O., Abualhaija, S., Torre, D., Sabetzadeh, M., & Briand, L. C. (2021). AI-enabled automation for completeness checking of privacy policies. *IEEE Transactions on Software Engineering*, *48*(11), 4647-4674.

Amor, R., & Dimyadi, J. (2021). The promise of automated compliance checking. *Developments in the Built Environment*, *5*, 100039. https://doi.org/https://doi.org/10.1016/j.dibe.2020.100039

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Badrah, M. K., MacLeod, I. A., & Kumar, B. (1998). Using object-communication for design standards modeling [Article]. *Journal of Computing in Civil Engineering*, *12*(3), 153-161. https://doi.org/10.1061/(ASCE)0887-3801(1998)12:3(153)

Barlow, H. B. (1989). Unsupervised learning. *Neural Computation*, *1*(3), 295-311.

Beach, T. H., Hippolyte, J.-L., & Rezgui, Y. (2020). Towards the adoption of automated regulatory compliance checking in the built environment. *Automation in Construction*, *118*, 103285. https://doi.org/https://doi.org/10.1016/j.autcon.2020.103285

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013). Towards automated compliance checking in the construction industry. Database and Expert Systems Applications: 24th International Conference, DEXA 2013, Prague, Czech Republic, August 26-29, 2013. Proceedings, Part I 24,

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013, 2013//). Towards Automated Compliance Checking in the Construction Industry. Database and Expert Systems Applications, Berlin, Heidelberg.

Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., & Thanh, N. L. (2012). Semantic web approach to ease regulation compliance checking in construction industry. *Future Internet*, *4*(3), 830-851.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., . . . Amodei, D. (2020). *Language Models are Few-Shot Learners* https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Cejas, O. A., Azeem, M. I., Abualhaija, S., & Briand, L. C. (2023). NLP-Based Automated Compliance Checking of Data Processing Agreements Against GDPR. *IEEE Transactions on Software Engineering*, *49*(9), 4282-4303. https://doi.org/10.1109/TSE.2023.3288901

Cesarotti, V., Benedetti, M., Dibisceglia, F., Di Fausto, D., Introna, V., La Bella, G., Martinelli, N., Ricci, M., Spada, C., & Varani, M. (2014). BIM–based approach to building operating management: a strategic lever to achieve efficiency, risk-shifting, innovation and sustainability. Proc. Conference: XVIII International Research Society for Public Management (IRSPM) Conference, at Ottawa, Canada,

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.* https://doi.org/10.1145/3641289

Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., Yang, L., Yi, X., Wang, C., & Wang, Y. (2023). A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Chen, N., Lin, X., Jiang, H., & An, Y. (2024). Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology. *Buildings*, *14*(7), 1983. https://www.mdpi.com/2075-5309/14/7/1983

Chen, Q., García de Soto, B., & Adey, B. T. (2018). Construction automation: Research areas, industry concerns and suggestions for advancement. *Automation in Construction*, *94*, 22-38. https://doi.org/https://doi.org/10.1016/j.autcon.2018.05.028

Chen, S. F., Beeferman, D., & Rosenfeld, R. (1998). Evaluation metrics for language models.

Chen, W., Du, X., Yang, F., Beyer, L., Zhai, X., Lin, T.-Y., Chen, H., Li, J., Song, X., & Wang, Z. (2021). A simple single-scale vision transformer for object localization and instance segmentation. *arXiv preprint arXiv:2112.09747*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, *23*(1), 155-162.

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval* (pp. 21-49). Springer.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*, 241-258.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words:

Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

El Naqa, I., & Murphy, M. J. (2015). *What is machine learning?* Springer.

Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., & Hemphill, L. (2023). A bibliometric review of large language models research from 2017 to 2023. *arXiv preprint arXiv:2304.02020*.

Fenves, S. J., Garrett, J., Kiliccote, H., Law, K., & Reed, K. (1995). Computer representations of design standards and building codes: US perspective. *The International Journal of Construction Information Technology*, *3*(1), 13-34.

Garrett James, H., & Fenves Steven, J. (1989). Knowledge-Based Standard-Independent Member Design. *Journal of Structural Engineering*, *115*(6), 1396-1411. https://doi.org/10.1061/(ASCE)0733-9445(1989)115:6(1396)

Garrett, J. H., & Fenves, S. J. (1987). A knowledge-based standards processor for structural component design. *Engineering with Computers*, *2*(4), 219-238. https://doi.org/10.1007/BF01276414

Ghahramani, Z. (2003). Unsupervised learning. In *Summer school on machine learning* (pp. 72-112). Springer.

Greenwood, D., Lockley, S., Malsane, S., & Matthews, J. (2010). Automated compliance checking using building information models. The Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors, Paris 2nd-3rd September,

Guo, D., Onstein, E., & Rosa, A. D. L. (2021). A Semantic Approach for Automated Rule Compliance Checking in Construction Industry. *IEEE Access*, *9*, 129648-129660. https://doi.org/10.1109/ACCESS.2021.3108226

Hao, Y., Qi, J., Ma, X., Wu, S., Liu, R., & Zhang, X. (2024). An LLM-Based Inventory Construction Framework of Urban Ground Collapse Events with Spatiotemporal Locations. *ISPRS International Journal of Geo-Information*, *13*(4), 133. https://www.mdpi.com/2220-9964/13/4/133

Hassani, S. (2024). Enhancing Legal Compliance and Regulation Analysis with Large Language Models. *arXiv preprint arXiv:2404.17522*.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ilal, S. M., & Günaydın, H. M. (2017). Computer representation of building codes for automated compliance checking. *Automation in Construction*, *82*, 43-58.

Iversen, O. (2024). *Leveraging Large Language Models for BIM-Based Automated Compliance Checking of Building Regulations* NTNU].

Joao, S.-J., Patricia, T., Juliana Parise, B., Barbara, P., Mike, K., Carlos Torres, F., & Julian, H. (2021). Automated compliance checking in healthcare building design. *Automation in Construction*, *129*, 103822. https://doi.org/https://doi.org/10.1016/j.autcon.2021.103822

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., & Tran-Johnson, E. (2022). Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., & Hüllermeier, E. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, *103*, 102274.

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of naacL-HLT,

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kruiper, R., Konstas, I., Gray, A., Sadeghineko, F., Watson, R., & Kumar, B. (2023). Document and query expansion for information retrieval on building regulations. Proc. 30th EG-ICE Int. Conf. Intell. Comput. Eng.,

Kruiper, R., Kumar, B., Watson, R., Sadeghineko, F., Gray, A., & Konstas, I. (2024). A platform-based Natural Language processing-driven strategy for digitalising regulatory compliance processes for the built environment. *Advanced Engineering Informatics*, *62*, 102653.

Kumar, B. (2012). Building Information Modeling: Road to 2016. *International Journal of 3-D Information Modeling (IJ3DIM)*, *1*(4), 1-7. https://doi.org/10.4018/ij3dim.2012100101

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. https://doi.org/10.1109/5.726791

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., & Rocktäschel, T. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459-9474.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., & Kumar, A. (2022). Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Liddy, E. D. (2001). Natural language processing.

Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., & Liu, Y. (2023). Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., & Ge, B. (2023). Summary of ChatGPT-Related research and perspective towards the future of large language models. *Meta-Radiology*, *1*(2), 100017. https://doi.org/https://doi.org/10.1016/j.metrad.2023.100017

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and*

Research (IJSR).[Internet], *9*(1), 381-386.

Malsane, S., Matthews, J., Lockley, S., Love, P. E. D., & Greenwood, D. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, *49*, 51-58. https://doi.org/https://doi.org/10.1016/j.autcon.2014.10.004

Martinelli, F., Mercaldo, F., Nardone, V., Orlando, A., Santone, A., & Vaglini, G. (2019). Model checking based approach for compliance checking [Article]. *Information Technology and Control*, *48*(2), 278-298. https://doi.org/10.5755/j01.itc.48.2.21724

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023). Prompt Engineering in Large Language Models. International Conference on Data Intelligence and Cognitive Informatics,

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, *27*(4), 12-12.

Mosbach, M., Pimentel, T., Ravfogel, S., Klakow, D., & Elazar, Y. (2023). Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation. *arXiv preprint arXiv:2305.16938*.

Muktadir, G. M. (2023). A Brief History of Prompt: Leveraging Language Models. *arXiv preprint arXiv:2310.04438*.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Ng, A., & Jordan, M. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*.

Nguyen, C., Bui, H., Nguyen, V., & Nguyen, T. (2023). An Approach to Generating API Test Scripts Using GPT. ACM International Conference Proceeding Series,

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., & Ray, A. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, *35*, 27730-27744.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),

Prabhakar, C., Li, H., Yang, J., Shit, S., Wiestler, B., & Menze, B. (2024). ViT-AE++: improving vision transformer autoencoder for self-supervised medical image representations. Medical Imaging with Deep Learning,

Pu, H., Yang, X., Li, J., & Guo, R. (2024). AutoRepo: A general framework for multimodal LLM-based automated construction reporting. *Expert Systems with Applications*, *255*, 124601. https://doi.org/https://doi.org/10.1016/j.eswa.2024.124601

Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. International Conference on Machine Learning,

Ratnayake, H., & Wang, C. (2024). A Prompting Framework to Enhance Language Model Output. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),

Reynolds, L., & McDonell, K. (2021). *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm* Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan. https://doi.org/10.1145/3411763.3451760

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536. https://doi.org/10.1038/323533a0

Sadeghineko, F., Kumar, B., & Chan, W. (2018, 2018//). A Semantic Web-Based Approach for Generating Parametric Models Using RDF. Advanced Computing Strategies for Engineering, Cham.

Sagar, M., Jane, M., Steve, L., Peter, E. D. L., & David, G. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, *49*, 51-58. https://doi.org/https://doi.org/10.1016/j.autcon.2014.10.004

Salama, D., & El-Gohary, N. (2011). Semantic modeling for automated compliance checking. In *Computing in Civil Engineering (2011)* (pp. 641-648).

Salama, D. A., & El-Gohary, N. M. (2013). Automated Compliance Checking of Construction Operation Plans Using a Deontology for the Construction Domain. *Journal of Computing in Civil Engineering*, *27*(6), 681-698. https://doi.org/doi:10.1061/(ASCE)CP.1943-5487.0000298

Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *44*(1.2), 206-226. https://doi.org/10.1147/rd.441.0206

Shahbaz, M., Suresh, L., Rexford, J., Feamster, N., Rottenstreich, O., & Hira, M. (2019). Elmo: Source routed multicast for public clouds. In *Proceedings of the ACM Special Interest Group on Data Communication* (pp. 458-471).

Sharma, P., & Yegneswaran, V. (2023). PROSPER: Extracting Protocol Specifications Using Large Language Models. HotNets 2023 - Proceedings of the 22nd ACM Workshop on Hot Topics in Networks,

Shin, J., Tang, C., Mohati, T., Nayebi, M., Wang, S., & Hemmati, H. (2023). Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated Software Engineering Tasks. *arXiv preprint arXiv:2310.10508*.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., . . . Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature*, *620*(7972), 172-180. https://doi.org/10.1038/s41586-023-06291-2

Soliman-Junior, J., Tzortzopoulos, P., Baldauf, J. P., Pedo, B., Kagioglou, M., Formoso, C. T., & Humphreys, J. (2021). Automated compliance checking in healthcare building design. *Automation in Construction*, *129*, 103822.

Song, H., Sun, D., Chun, S., Jampani, V., Han, D., Heo, B., Kim, W., & Yang, M.-H. (2021). Vidt: An efficient and effective fully transformer-based object detector. *arXiv preprint arXiv:2110.03921*.

Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. Proceedings of the IEEE/CVF international conference on computer vision,

Teubner, T., Flath, C. M., Weinhardt, C., van der Aalst, W., & Hinz, O. (2023). Welcome to the era of chatgpt et al. the prospects of large language models. *Business & Information Systems Engineering*, *65*(2), 95-101.

Titus, L. M. (2024). Does ChatGPT have semantic understanding? A problem with the statistics-of-occurrence strategy. *Cognitive Systems Research*, *83*, 101174. https://doi.org/https://doi.org/10.1016/j.cogsys.2023.101174

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., & Azhar, F. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., & Bhosale, S. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, *109*(2), 373-440.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017a). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017b). *Attention is All you Need* https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Voetman, R., van Meekeren, A., Aghaei, M., & Dijkstra, K. (2023). Using Diffusion Models for Dataset Generation: Prompt Engineering vs. Fine-Tuning. In N. Tsapatsoulis, A. Lanitis, M. Pattichis, C. Pattichis, C. Kyrkou, E. Kyriacou, Z. Theodosiou, & A. Panayides, *Computer Analysis of Images and Patterns* Cham.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, *12*(3), 729.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine learning*, *38*, 257-286.

Wu, S., & Manber, U. (1992). Fast text searching: allowing errors. *Communications of the ACM*, *35*(10), 83-91.

Wu, Y.-c., & Feng, J.-w. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, *102*, 1645-1656.

Xu, F. F., Alon, U., Neubig, G., & Hellendoorn, V. J. (2022). A systematic evaluation of large language models of code. Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming,

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press.

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering*, *30*(2), 04015014.

Zhang, R., & El-Gohary, N. (2021). A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Automation in Construction*, *132*, 103834. https://doi.org/https://doi.org/10.1016/j.autcon.2021.103834

Zhang, Y., & Yang, Q. (2022). A Survey on Multi-Task Learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), 5586-5609. https://doi.org/10.1109/TKDE.2021.3070203

Zhang, Z., Ma, L., & Broyd, T. (2023). Rule capture of automated compliance checking of building requirements: a review. *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, *40*(XXXX), 1-15.

Zhang, Z., Nisbet, N., Ma, L., & Broyd, T. (2023). Capabilities of rule representations for automated compliance checking in healthcare buildings. *Automation in Construction*, *146*, 104688. https://doi.org/https://doi.org/10.1016/j.autcon.2022.104688

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., & Du, M. (2024). Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, *15*(2), 1-38.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., & Dong, Z. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zheng, Z., Zhou, Y.-C., Lu, X.-Z., & Lin, J.-R. (2022). Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. *Automation in*

Construction, *142*, 104524. https://doi.org/https://doi.org/10.1016/j.autcon.2022.104524

Zhou, Y.-C., Zheng, Z., Lin, J.-R., & Lu, X.-Z. (2022). Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry*, *142*, 103746. https://doi.org/https://doi.org/10.1016/j.compind.2022.103746

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., & Anadkat, S. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Amaral, O., Abualhaija, S., Torre, D., Sabetzadeh, M., & Briand, L. C. (2021). AI-enabled automation for completeness checking of privacy policies. *IEEE Transactions on Software Engineering*, *48*(11), 4647-4674.

Amor, R., & Dimyadi, J. (2021). The promise of automated compliance checking. *Developments in the Built Environment*, *5*, 100039. https://doi.org/https://doi.org/10.1016/j.dibe.2020.100039

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Barlow, H. B. (1989). Unsupervised learning. *Neural Computation*, *1*(3), 295-311.

Beach, T. H., Hippolyte, J.-L., & Rezgui, Y. (2020). Towards the adoption of automated regulatory compliance checking in the built environment. *Automation in Construction*, *118*, 103285. https://doi.org/https://doi.org/10.1016/j.autcon.2020.103285

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013). Towards automated compliance checking in the construction industry. Database and Expert Systems Applications: 24th International Conference, DEXA 2013, Prague, Czech Republic, August 26-29, 2013. Proceedings, Part I 24,

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013, 2013//). Towards Automated Compliance Checking in the Construction Industry. Database and Expert Systems Applications, Berlin, Heidelberg.

Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., & Thanh, N. L. (2012). Semantic web approach to ease regulation compliance checking in construction industry. *Future Internet*, *4*(3), 830-851.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., . . . Amodei, D. (2020). *Language Models are Few-Shot Learners* https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Cejas, O. A., Azeem, M. I., Abualhaija, S., & Briand, L. C. (2023). NLP-Based Automated

Compliance Checking of Data Processing Agreements Against GDPR. *IEEE Transactions on Software Engineering*, *49*(9), 4282-4303. https://doi.org/10.1109/TSE.2023.3288901

Cesarotti, V., Benedetti, M., Dibisceglia, F., Di Fausto, D., Introna, V., La Bella, G., Martinelli, N., Ricci, M., Spada, C., & Varani, M. (2014). BIM–based approach to building operating management: a strategic lever to achieve efficiency, risk-shifting, innovation and sustainability. Proc. Conference: XVIII International Research Society for Public Management (IRSPM) Conference, at Ottawa, Canada,

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.* https://doi.org/10.1145/3641289

Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., Yang, L., Yi, X., Wang, C., & Wang, Y. (2023). A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Chen, Q., García de Soto, B., & Adey, B. T. (2018). Construction automation: Research areas, industry concerns and suggestions for advancement. *Automation in Construction*, *94*, 22-38. https://doi.org/https://doi.org/10.1016/j.autcon.2018.05.028

Chen, S. F., Beeferman, D., & Rosenfeld, R. (1998). Evaluation metrics for language models.

Chen, W., Du, X., Yang, F., Beyer, L., Zhai, X., Lin, T.-Y., Chen, H., Li, J., Song, X., & Wang, Z. (2021). A simple single-scale vision transformer for object localization and instance segmentation. *arXiv preprint arXiv:2112.09747*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, *23*(1), 155-162.

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval* (pp. 21-49). Springer.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*, 241-258.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

El Naqa, I., & Murphy, M. J. (2015). *What is machine learning?* Springer.

Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., & Hemphill, L. (2023). A bibliometric review of large language models research from 2017 to 2023. *arXiv preprint arXiv:2304.02020*.

Fenves, S. J., Garrett, J., Kiliccote, H., Law, K., & Reed, K. (1995). Computer representations of design standards and building codes: US perspective. *The International Journal of Construction Information Technology*, *3*(1), 13-34.

Garrett James, H., & Fenves Steven, J. (1989). Knowledge-Based Standard-Independent Member Design. *Journal of Structural Engineering*, *115*(6), 1396-1411. https://doi.org/10.1061/(ASCE)0733-9445(1989)115:6(1396)

Garrett, J. H., & Fenves, S. J. (1987). A knowledge-based standards processor for structural component design. *Engineering with Computers*, *2*(4), 219-238. https://doi.org/10.1007/BF01276414

Ghahramani, Z. (2003). Unsupervised learning. In *Summer school on machine learning* (pp. 72-112). Springer.

Greenwood, D., Lockley, S., Malsane, S., & Matthews, J. (2010). Automated compliance checking using building information models. The Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors, Paris 2nd-3rd September,

Guo, D., Onstein, E., & Rosa, A. D. L. (2021). A Semantic Approach for Automated Rule Compliance Checking in Construction Industry. *IEEE Access*, *9*, 129648-129660. https://doi.org/10.1109/ACCESS.2021.3108226

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ilal, S. M., & Günaydın, H. M. (2017). Computer representation of building codes for automated compliance checking. *Automation in Construction*, *82*, 43-58.

Joao, S.-J., Patricia, T., Juliana Parise, B., Barbara, P., Mike, K., Carlos Torres, F., & Julian, H. (2021). Automated compliance checking in healthcare building design. *Automation in Construction*, *129*, 103822. https://doi.org/https://doi.org/10.1016/j.autcon.2021.103822

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., & Tran-Johnson, E. (2022). Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of naacL-HLT,

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kruiper, R., Konstas, I., Gray, A., Sadeghineko, F., Watson, R., & Kumar, B. (2023). Document and query expansion for information retrieval on building regulations. Proc. 30th EG-ICE Int. Conf. Intell. Comput. Eng.,

Kruiper, R., Kumar, B., Watson, R., Sadeghineko, F., Gray, A., & Konstas, I. (2024). A platform-based Natural Language processing-driven strategy for digitalising regulatory compliance processes for the built environment. *Advanced Engineering Informatics*, *62*, 102653.

Kumar, B. (2012). Building Information Modeling: Road to 2016. *International Journal of 3-D Information Modeling (IJ3DIM)*, *1*(4), 1-7. https://doi.org/10.4018/ij3dim.2012100101

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. https://doi.org/10.1109/5.726791

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., & Rocktäschel, T. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459-9474.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., & Kumar, A. (2022). Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Liddy, E. D. (2001). Natural language processing.

Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., & Liu, Y. (2023). Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., & Ge, B. (2023). Summary of ChatGPT-Related research and perspective towards the future of large language models. *Meta-Radiology*, *1*(2), 100017. https://doi.org/https://doi.org/10.1016/j.metrad.2023.100017

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), 381-386.

Malsane, S., Matthews, J., Lockley, S., Love, P. E. D., & Greenwood, D. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, *49*, 51-58. https://doi.org/https://doi.org/10.1016/j.autcon.2014.10.004

Martinelli, F., Mercaldo, F., Nardone, V., Orlando, A., Santone, A., & Vaglini, G. (2019). Model checking based approach for compliance checking [Article]. *Information Technology and Control*, *48*(2), 278-298. https://doi.org/10.5755/j01.itc.48.2.21724

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023). Prompt Engineering in Large Language Models. International Conference on Data Intelligence and Cognitive Informatics,

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, *27*(4), 12-12.

Mosbach, M., Pimentel, T., Ravfogel, S., Klakow, D., & Elazar, Y. (2023). Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation. *arXiv preprint arXiv:2305.16938*.

Muktadir, G. M. (2023). A Brief History of Prompt: Leveraging Language Models. *arXiv preprint* *arXiv:2310.04438*.

Ng, A., & Jordan, M. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*.

Nguyen, C., Bui, H., Nguyen, V., & Nguyen, T. (2023). An Approach to Generating API Test Scripts Using GPT. ACM International Conference Proceeding Series,

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., & Ray, A. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, *35*, 27730-27744.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),

Prabhakar, C., Li, H., Yang, J., Shit, S., Wiestler, B., & Menze, B. (2024). ViT-AE++: improving vision transformer autoencoder for self-supervised medical image representations. Medical Imaging with Deep Learning,

Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. International Conference on Machine Learning,

Ratnayake, H., & Wang, C. (2024). A Prompting Framework to Enhance Language Model Output. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),

Reynolds, L., & McDonell, K. (2021). *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm* Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan. https://doi.org/10.1145/3411763.3451760

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536. https://doi.org/10.1038/323533a0

Sagar, M., Jane, M., Steve, L., Peter, E. D. L., & David, G. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, *49*, 51-58. https://doi.org/https://doi.org/10.1016/j.autcon.2014.10.004

Salama, D., & El-Gohary, N. (2011). Semantic modeling for automated compliance checking. In *Computing in Civil Engineering (2011)* (pp. 641-648).

Salama, D. A., & El-Gohary, N. M. (2013). Automated Compliance Checking of Construction Operation Plans Using a Deontology for the Construction Domain. *Journal of*

Computing in Civil Engineering, *27*(6), 681-698. https://doi.org/doi:10.1061/(ASCE)CP.1943-5487.0000298

Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *44*(1.2), 206-226. https://doi.org/10.1147/rd.441.0206

Shahbaz, M., Suresh, L., Rexford, J., Feamster, N., Rottenstreich, O., & Hira, M. (2019). Elmo: Source routed multicast for public clouds. In *Proceedings of the ACM Special Interest Group on Data Communication* (pp. 458-471).

Sharma, P., & Yegneswaran, V. (2023). PROSPER: Extracting Protocol Specifications Using Large Language Models. HotNets 2023 - Proceedings of the 22nd ACM Workshop on Hot Topics in Networks,

Shin, J., Tang, C., Mohati, T., Nayebi, M., Wang, S., & Hemmati, H. (2023). Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated Software Engineering Tasks. *arXiv preprint arXiv:2310.10508*.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., . . . Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature*, *620*(7972), 172-180. https://doi.org/10.1038/s41586-023-06291-2

Soliman-Junior, J., Tzortzopoulos, P., Baldauf, J. P., Pedo, B., Kagioglou, M., Formoso, C. T., & Humphreys, J. (2021). Automated compliance checking in healthcare building design. *Automation in Construction*, *129*, 103822.

Song, H., Sun, D., Chun, S., Jampani, V., Han, D., Heo, B., Kim, W., & Yang, M.-H. (2021). Vidt: An efficient and effective fully transformer-based object detector. *arXiv preprint arXiv:2110.03921*.

Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. Proceedings of the IEEE/CVF international conference on computer vision,

Titus, L. M. (2024). Does ChatGPT have semantic understanding? A problem with the statistics-of-occurrence strategy. *Cognitive Systems Research*, *83*, 101174. https://doi.org/https://doi.org/10.1016/j.cogsys.2023.101174

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., & Azhar, F. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., & Bhosale, S. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, *109*(2), 373-440.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017a). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017b). *Attention is All you Need* https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Voetman, R., van Meekeren, A., Aghaei, M., & Dijkstra, K. (2023). Using Diffusion Models for Dataset Generation: Prompt Engineering vs. Fine-Tuning. In N. Tsapatsoulis, A. Lanitis, M. Pattichis, C. Pattichis, C. Kyrkou, E. Kyriacou, Z. Theodosiou, & A. Panayides, *Computer Analysis of Images and Patterns* Cham.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, *12*(3), 729.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine learning*, *38*, 257-286.

Wu, S., & Manber, U. (1992). Fast text searching: allowing errors. *Communications of the ACM*, *35*(10), 83-91.

Wu, Y.-c., & Feng, J.-w. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, *102*, 1645-1656.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press.

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering*, *30*(2), 04015014. https://doi.org/doi:10.1061/(ASCE)CP.1943-5487.0000346

Zhang, R., & El-Gohary, N. (2021). A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Automation in Construction*, *132*, 103834. https://doi.org/https://doi.org/10.1016/j.autcon.2021.103834

Zhang, Y., & Yang, Q. (2022). A Survey on Multi-Task Learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), 5586-5609. https://doi.org/10.1109/TKDE.2021.3070203

Zhang, Z., Ma, L., & Broyd, T. (2023). Rule capture of automated compliance checking of building requirements: a review. *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, *40*(XXXX), 1-15.

Zhang, Z., Nisbet, N., Ma, L., & Broyd, T. (2023). Capabilities of rule representations for automated compliance checking in healthcare buildings. *Automation in Construction*, *146*, 104688. https://doi.org/https://doi.org/10.1016/j.autcon.2022.104688

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., & Du, M. (2024). Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, *15*(2), 1-38.

Zheng, Z., Zhou, Y.-C., Lu, X.-Z., & Lin, J.-R. (2022). Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. *Automation in Construction*, *142*, 104524. https://doi.org/https://doi.org/10.1016/j.autcon.2022.104524

Zhou, Y.-C., Zheng, Z., Lin, J.-R., & Lu, X.-Z. (2022). Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry*, *142*, 103746. https://doi.org/https://doi.org/10.1016/j.compind.2022.103746