A GPT-based method of Automatic Compliance Checking through prompt engineering

Xiaoyu Liu, Haijiang Li*, Xiaofeng Zhu
Cardiff University, United Kingdom
LiuX133@cardiff.ac.uk

Abstract: Compliance-checking is critical in the Architecture, Engineering and Construction (AEC) industry as it ensures that a project adheres to relevant codes and regulations. However, typical compliance-checking methodologies rely heavily on manual labour, which could be time-consuming and costly. Furthermore, the uncertainty of errors and redundancy can cause resources wastes and potential risks for the project. Digitalisation has paved the way for the development of automatic data processing. Hence, the recent research development in Deep learning (DL) and Large Language Models (LLMs) make it possible for machines to understand information from documents, including regulations and implementation documents from relevant projects. This research introduces a method to implement Automatic Compliance Checking (ACC) process to building design specifications, organizes 4 tests to evaluate GPT-based models, and proposes suggestions about improving the performance of LLMs through prompt engineering.

1. Introduction:

1.1 Development progression of digitalization in the AEC industry

With the rapid development of Digital Technologies (DT) in the AEC industry, on the one hand, the abilities of information processing and coordinating among various parties are significantly improved (Manzoor et al., 2021; Wang et al., 2022). However, on the other hand, the cost has risen sharply due to the need for humans to process a massive amount of data (García de Soto et al., 2022). Consequently, there is a growing trend towards deploying automated information processing technologies on BIM-based models to replace a significant amount of repetitive information processing work. The AEC industry seeks to digitize the process of project execution as much as possible, with one significant reason being the ability to have machines process and analyze vast amounts of raw data from project information, which is then submitted to humans for decision-making and planning (Ren et al., 2022; Zhu & Augenbroe, 2006). A significant amount of research has been dedicated to automated compliance checking for BIM models, and considerable progress has been made in this area. Many checks for BIM models can assist engineers in making decisions and developing plans to a considerable extent (Beach et al., 2013; Dimyadi & Amor, 2013). With the flourishing of the software engineering industry, automatic checks for BIM models, such as Revit models, have been commercialized to a significant extent. This has allowed engineers and designers to incorporate safety, environmental, and related standards into the design consideration during the early stages of the project. However, there is still potential for processing the vast amount of textual content produced in projects.

1.2 Development progression of Natural Language Processing (NLP) in the AEC industry

NLP research has made significant contributions to the understanding of standards. For example, Hassan & Le (2023) utilized word embedding techniques (Word2Vec) to convert unstructured textual data into structured data that can be processed by machines (Hassan & Le, 2023). They then applied an ontology-based approach to identify semantic features in standards, classify requirements, and prioritize them, thereby facilitating rule checking. This approach has been successful in classifying the content of legal standards, in other words, it can match content with similar semantics. There are many types of research focusing on the downstream applications of multi-classification to text through Large Language Models, and this research provides significant performance in some specific tasks. Though these applications are still limited to the comprehension of in-context information, there is no doubt that these researches have driven the development of LLMs processing ARC.

Fortunately, the emergence of DL-based NLP techniques appears to offer a way to learn from file contents to a certain extent (Albawi et al., 2018), although further research is needed to confirm this claim. Conventional NLP techniques based on CNN have achieved basic functions such as text classification (Haitao et al., 2020), translation (Chen & Wu, 2017), sentiment analysis(Indhraom Prabha & Umarani Srikanth, 2019), and automatic question-answering. However, achieving high accuracy with CNN-based NLP models generally requires a vast amount of training data and more iterations, which can be costly for many research teams, organizations, and small businesses. The situation was not significantly improved until the release of the Transformer model from Google in 2017 (Vaswani et al., 2017). In 2018, pre-trained models based on the Transformer architecture were developed, greatly reducing the research costs of exploring DL models. The Transformer framework has been one of the most popular frameworks in the NLP field in recent years, and almost all DL models for NLP have been developed based on the Transformer after its publication (Kalyan et al., 2021). This is not to say that models based on the Transformer are not complex; on the contrary, pre-trained models based on the Transformer are among the largest, most expensive to train, and most complex models to date.

1.3 The state-of-art NLP technologies

Taking the most representative GPT series from OpenAI (Brown et al., 2020; Ouyang et al., 2022; Radford et al., 2018; Radford et al., 2019) and BERT from Google (Devlin et al., 2018) as examples, these two deep pre-trained models have made it possible to deploy DL models with NLP capabilities in various fields. Prior to the release of ChatGPT, BERT had been the most popular model in academia, not only because of its impressive performance on datasets for various tasks but also because of its versatility to handle most NLP tasks from various industries. However, at the end of 2022, the InstructGPT model emerged as the hottest AI engine due to its powerful natural language processing capabilities following the launch of ChatGPT.

As today's Large Language models architecture, high-quality logical generation generally requires the construction of complex Neural Networks (NNs) models, large datasets, and extended periods of training and debugging. The associated costs are often in the tens of millions, which creates significant pressure for institutions and companies. This problem persisted until the emergence of pre-trained models. However, research has shown that ChatGPT has instability issues in text generation tasks. For example, it can learn the patterns and content of prompt examples, outputting similar corresponding content, but the reasoning behind it is often lacking or fabricated. This raises questions about ChatGPT's behaviour in

some creative text generation problems, such as whether design explanations generated without comprehensive human review are genuine. However, in other tasks, ChatGPT can produce relatively satisfactory results in reasoning between prompts and completions, such as text translation and emotion recognition. Based on this understanding, we further reason whether ChatGPT can check the validity of the corresponding design explanation texts based on architectural design regulations and generate logical reasoning between the design explanation and regulations. If there is doubt about the validity of the reasoning, we can require ChatGPT to provide an explanation of its judgment to verify its rationale. Based on this perspective, this study proposes a method for prompt learning and generation of design explanations based on ChatGPT and automated compliance checks of design explanation texts against regulations.

1.4 Paper distribution

Chapter 1 of this paper introduces the research background, including the digitization process of architectural design in research and the necessity of automated data processing. Chapter 2 outlines the assumptions made for this research and limits the scope of the corresponding experiments. Chapter 3 elaborates on the methodology of the research, explaining the principles and functions of GPT models and the expected effects of this study. Chapter 4 explains the experimental process and presents the experimental results, including model fine-tuning, selection, and prompt construction, as well as the construction of the training set used for fine-tuning. Chapter 5 analyzes and discusses the experimental results, evaluates the achievements of this research and provides prospects for future work.

2. Hypothesis

Before providing a detailed explanation of the research methodology, essential hypotheses are provided.

2.1 Overall framework for the entire automated compliance checking system.

This framework consists of three components: retrieve, match, and check.

Retrieve: This involves identifying and extracting independent sections of the design specification for checking. Due to limitations in length, the review of related content, such as drawings, information models, and execution process files, is not presented in this paper.

Match: This component involves matching the design specification with corresponding regulations or standards. As there may be multiple requirements for the design content, it is necessary to list them all within the scope of the regulations.

Check: This involves comprehending the content of the design specification and its corresponding regulations to automatically determine whether the design specification complies with the corresponding regulations.

After performing the above steps, an automated compliance check is considered to be fully executed. Previous studies have shown that the retrieve and match functions can be achieved through existing methods. Therefore, the next step is to verify the language model's understanding of the design specification and regulations.

2.2 Large Language Models

According to Liang et al. research, GPT series models provide extraordinary performance in most scenarios(Liang et al., 2022). Consequently, the capability of the models can be assumed to meet the required scenarios if the prompts are generated appropriately. Though there are several types of LLMs produced to address different scenarios, in that way, both prompts and models need to be organized and tested for the ARC scenario. Finally, the research proposes a framework for how to design prompts for different scenarios.

3. Methodology

3.1 General design

This research uses the GPT-3 series of models, which is a natural language generation model based on transformer architecture. By using the prompt approach, the model generates compliance check results by learning the internal logic connection between regulations and specifications within the prompt.

Figure 1 presents the general design of implementing research. According to the architecture of the transformer, the basic scenario of the GPT-based models could be considered as sentence prediction from context. So the first step to drive the LLMs is generating appropriate prompts based on the target scenario. In general, there are 3 types of prompts for LLMs, zero-shot learning, one-shot learning, and few shots learning(Saravia, 2022), the details of the prompt design are explained in Session 4. Experiment. When the prompts have been prepared, they are fed in LLMs separately in 2 different ways, one way is directly feeding in complete models, and the other way is applying fine-tuning process before the test. Both of the models produce completions (or results) based on a prompt. According to the qualities of completions, the models' performance can be assessed. Furthermore, both performance of prompt and LLMs can be analyzed through variables controlled during the experiment. Finally, according to the analyses of the prompts and models, the prompts can be modified, and correspondingly the models with the best performance are applied for further research.

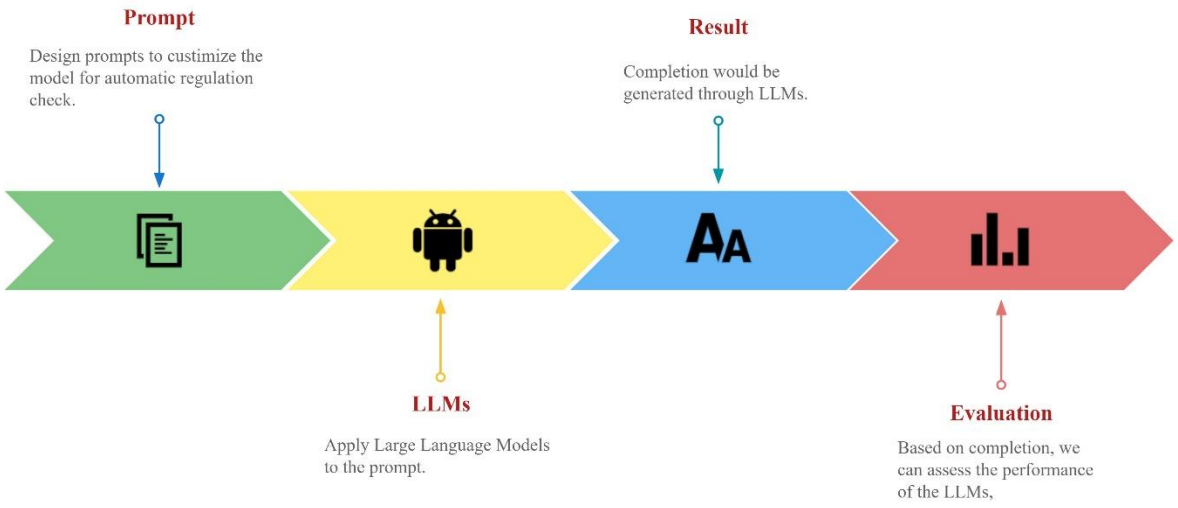

Figure 1, the design of the research implementation.

Figure 2 presents the detailed processing procedure for realizing LLMs-based ACC through prompt engineering. According to the figure, 2 types of models are applied, GPT-3-based fine-tuned models and GPT-3.5-based complete models. As reinforcement learning methods are integrated, GPT-3.5-based models present a better performance in multi-tasks. Consequently, they can realize deep analysis to large complex (maximum 4096 tokens in GPT-3.5 based models) contexts without fine-tuning. However, GPT-3 base models don't have such powerful language processing performance in general, but the models can be boosted through the fine-tuning process to reach the same level of performance.

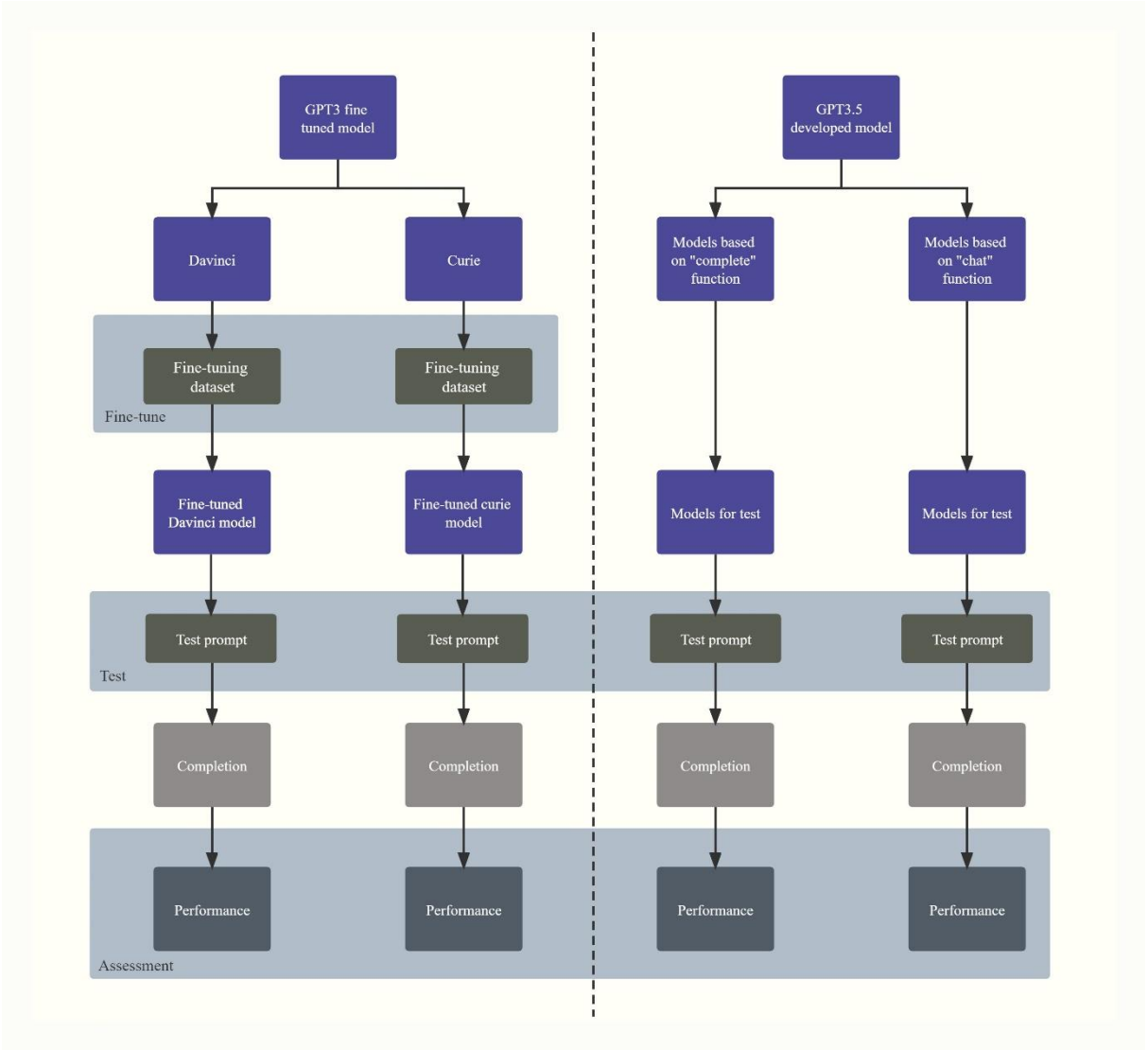

Figure 2, the flow chart of automatic compliance checks through LLMs

For the two main types of models:

GPT-3 based models: The dataset of prompt includes hundreds of samples with organised structure pairs: "prompt with samples of instructions, building design regulations' clause, and completion with samples of checking results".

GPT-3.5 prepared models: This type of model is capable of implementing the tasks proposed through prompt, which provides detailed instructions, and several precise samples to limit the form of generated results.

As both types of models complete essential preparation, their performance of the specific task can be evaluated through unified prompt tests. There are 2 different ways to evaluate the models from different perspectives through prompt designing.

The first is to build simple, organized, structured prompts which only have only one sentence of the instruction following no example or few examples. This task evaluates LLMs' basic capacities of learning from context and tries to figure out the boosting capacities of fine-tuning process. Consequently, the organized structure of the test prompts are the same as the fine-tuning prompt. This task is assumed to be completed by most of the tasks hence batch processing is implemented in this task. The generated results are divided into 2 types, and accordingly, a confusion matrix is built. Finally, a comprehensive quantitative performance evaluation in this scenario is provided.

The second is to build large, complex, naturally structured prompts which have a more similar form to general design documents. This task generally requires LLMs to learn more complicated internal logical connections from instructions and examples and applies them to completion generating. The capacities of LLMs can be claimed if the task can be realized precisely.

There are 2 types of experimental forms in this research, for complex large, naturally structured contexts, the "playground" from OpenAI official website is applied for the test environment. And API of ChatGPT is applied for batch processing in Python. The generated results are recorded in an extra column of original data to be converted into a confusion matrix. Due to space limitations, all of the codes, datasets and row data of the experimental results can be found at the following link: https://github.com/xiaoyuliu822/GPT-based-ACC.git.

4. Experiment

4.1 LLMs distribution

In this research, 7 GPT-based LLMs are applied for the ACC scenario, and the performance of the models is evaluated. Figure 3 presents the distribution of the tested GPT-based LLMs. There are 2 types of essential scenarios for GPT-based LLMs, Chat and Completion.

- The chat model can be recognized as a simplified version of the text-davinci-003 completion model, this model tries to provide the same performance as the most capable model but save 99% on calculation cost.
- The other types of models are typical completion models, which have subclasses of fine-tuned and prepared models. As the GPT-3 based models have the most capacities, Cuire and Davnici are applied for fine-tuning before being tested with unified prompts. All of the latest GPT-3.5-based prepared models are tested, they are text-davinci-003, text-curie-001, text-babbge-001, and text-ada-001 in the order of capacities.

4.2 Prompt engineering

In this subsection, the details of the prompt design are explained comprehensively, including tasks, structures, and contents. As previous introducing, tasks include fine-tuning and unified tests, structures include natural and organised 2 types, and contents are extracted from HTM 05-02 fire safety codes.

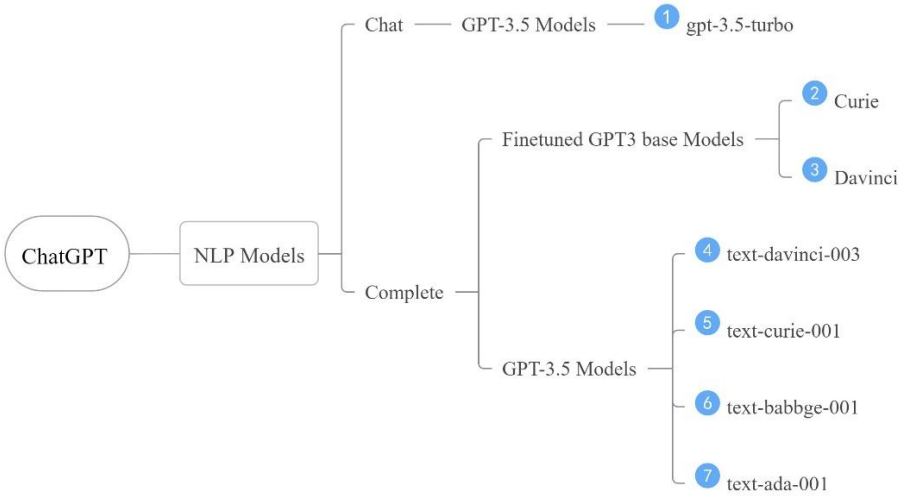

Figure 3: Test models based on ChatGPT

4.2.1 Prompt for fine-tuning

In fine-tuning task, prompts are built into a dataset which contains hundreds of independent samples, Figure 4 presents the fine-tuning prompt structure of the dataset. These samples comprise all of the examples of compliance checking to build design specifications under HTM 05-02 fire safety codes. This process is designed to enhance the capability of GPT-3 based models learning, inference and judgment during implementing compliance checks. The dataset is divided into the fine-tuning set, validating set, and testing set in a ratio of 8:1:1.

Fine-tuned models are pre-tested on validating set before the unified test. Figure 5 presents the fine-tuned models' performance on validating set. There is a confusion matrix and 4 indices for each of the fine-tuned models. According to the figure, there are 3 basic statuses in this task, "negative" means "the requirement is not met", "positive" means "the requirement is met", and "task fail" means "the model produces meaningless results or the model doesn't understand the task". Each row of the matrix represents the statuses of the true value which are provided by the dataset, there are only 2 rows, "negative" and "positive", as no meaningless tasks are provided, which can be seen through the vertical axis. Each column of the matrix represents the statuses of the predicted value generated by models, the same values as predictions.

Four main evaluation indices are calculated in validating process, and Chart 1 explains each element of a confusion matrix:

Accuracy: the proportion of correct predictions out of the total predictions:

$$A = \frac{(TP+TN)}{N}.$$

Precision: the proportion of true positives ("the requirement is met") out of all the positive predictions ("the requirement is met"):

$$P = \frac{TP}{(TP+FP)}.$$

Recall: the proportion of true positives out of all the actual positive values:

$$R = \frac{TP}{(TP+FN)}.$$

F1 score: the harmonic mean of precision and recall, it is useful when dealing with imbalanced datasets:

$F1 = \frac{2}{(\frac{1}{P}+\frac{1}{R})}$ (Goutte & Gaussier, 2005).

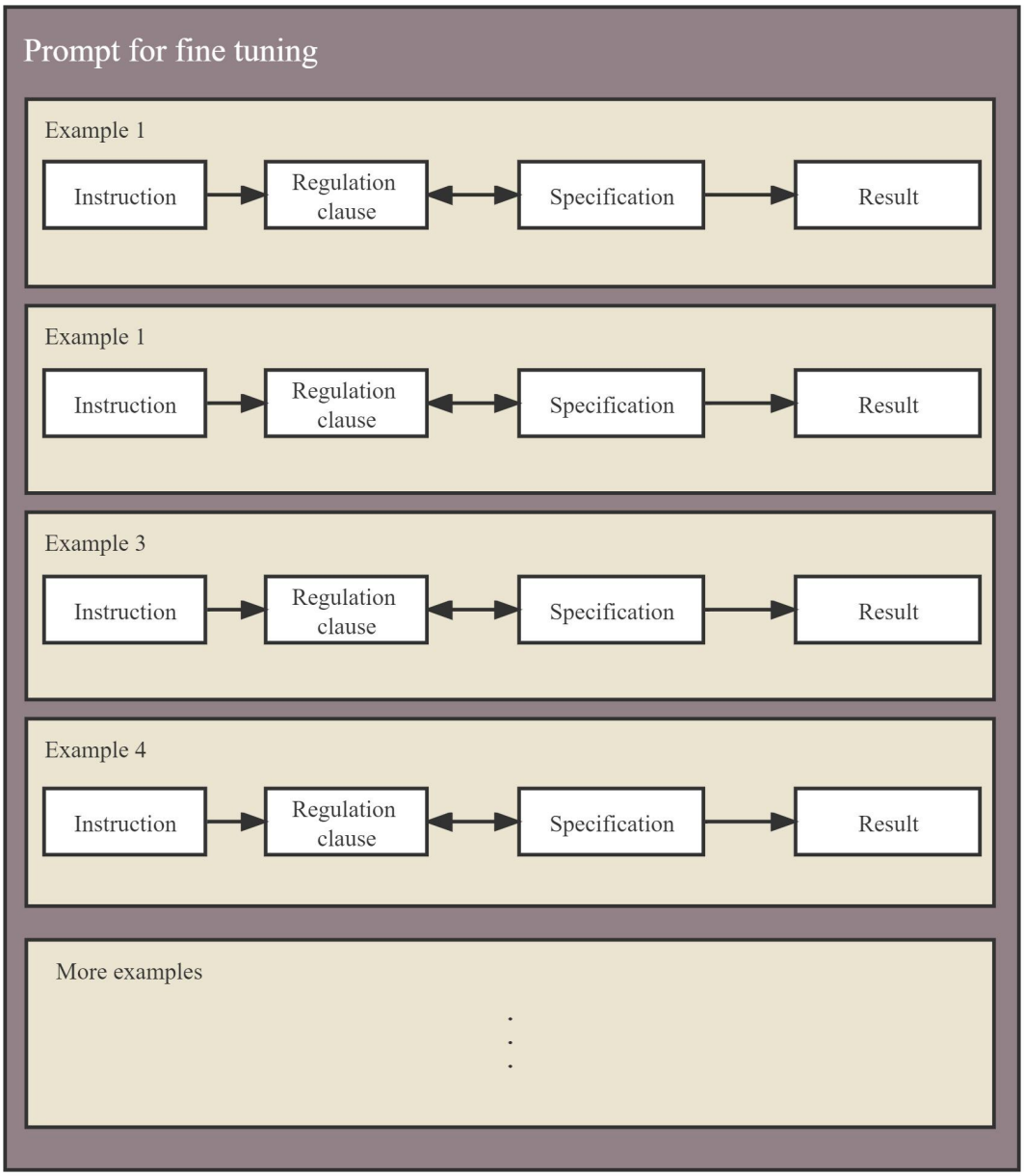

Figure 4. The prompt design for fine-tuning.

TN: True Negative, the true values are negative, and the predicted values are negative.

FN: False Negative, the true values are negative, and the predicted values are positive.

FP: False Positive, the true values are positive, and the predicted values are negative.

TP: True Positive, the true values are positive, and the predicted values are positive.

TF: Task failure, the tasks failed during implementation.

TN01	FN02	TF03
FP11	TP12	TF13
TF21	TF22	TF23

Table 1 Confusion matrix.

The performance of 2 fine-tuned models is presented in Figure 5. According to the figure, there are 21 samples in the validating dataset, 14 of them are positive, and 7 are negative, fine-tuned curie model has better performance than fine-tuned davinci model in this dataset, and both of the models provide acceptable performance. The fine-tuned models are prepared after the validating process is complete, and they are provided for further unified tests with other GPT-3.5 based models.

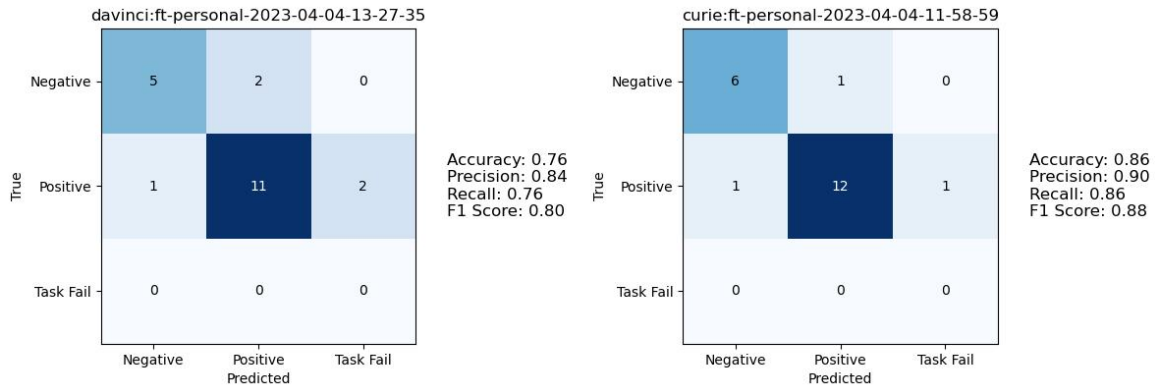

Figure 5. The confusion matrices of fine-tuning models on the validation dataset.

4.2.2 Prompt for testing

In the testing task, 4 prompts in 2 types are built for different capabilities, as figure 6 presenting, the first 2 prompts adopt an organized structure context, and the second 2 prompts adopt a naturally structured context.

Test 1: test 1 is the most simplified prompt in this research, it adopts the same organized structure as the prompts in fine-tuning process, the prompt is divided into several parts with symbolled separators, and completion is required to produce results from LLMs, no examples are given, true results are provided for comparison after generating. The LLMs need to produce results directly. This prompt design is applied mainly for quantifying models' performance. Through a series of evaluation indices, including accuracy, precision, recall, F1 score and confusion matrices, the model's performance can be precisely evaluated and visualized. This would directly prove that the GPT-3 models' capability can be boosted closely to a GPT-3.5 model.

Test 2: the prompt of test 2 adopts a similar organized structure, this is an extension of test 1, in which all the symbolled separators are cancelled in prompts and examples are provided to boost the generalization of the tested models instead. The examples are built as clauses pairs

of fire safety regulations and building design specifications, the true results are given for learning. The test clauses pair from regulations and specifications are provided with the following, which requires LLMs to produce the results based on their learning results. The organised structures of the test prompts are designed to evaluate the GPT-3 based models considering the models have less capacity compared with the GPT-3.5 models. The capability of retrieving is not integrated directly into the GPT-3 models. Hence pre-processing operations are required before implementing the compliance check.

Test 3: the prompt of test 3 adopts a natural structure, though, in this test, no examples are given. The regulation clauses and corresponding building design specifications are listed separately, and the LLMs are required to produce the results directly. This task is designed to simulate the general ACC process in the deployment environment, the LLMs should match every regulation clause pair from whole documents of regulations and specifications, then implement the checking process and produce the results.

Test 4: this prompt of test 4 adopts a natural structure, which means the contexts are designed closely to the general documents. The instructions are detailed and precise to describe the scope of compliance checking, in this research, several fire safety regulation clauses and corresponding building design specifications are listed together in prompts as examples which helps LLMs to learn the compliance checking process from internal logic connections within the clauses and specifications pairs. The specified forms of generated completions can help LLMs to provide required results representations like "0" stands for false, and "1" for true. Finally, 2 test building design specifications are given to let the LLMs implement the ACC process. In this test, the LLMs should match the regulation clauses with building design specifications first, then implement the checking process and produce the results.

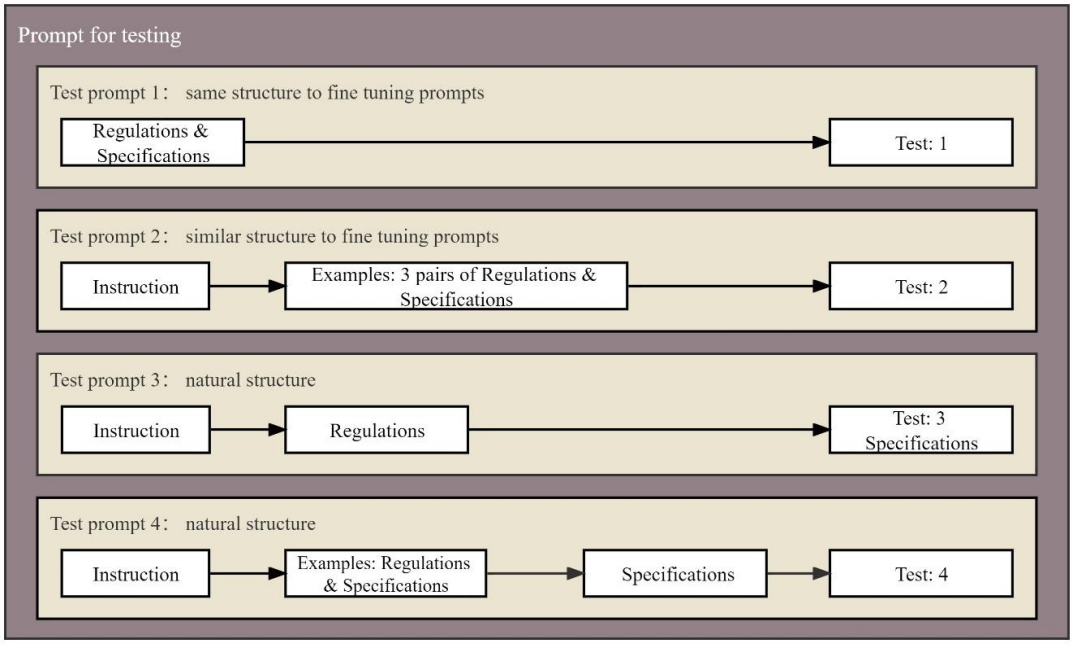

Figure 6. The prompt for testing.

4.3 Contents

In this research, clauses of fire safety regulations and responding building design specifications. Fire safety codes are extracted from *Health Technical Memorandum 05-02:*

Firecode, fire safety in the design of Healthcare premises (HTM 05-02) (Health, 2015). The responding building design specifications are generated by the author based on general design principles from various virtual projects.

4.4 Results

4.4.1 General results

The initial results of the experiments are recorded in Table 2. This is a general review of the tests' results, in the chart, "0" means the model fails the test, "1" means the model passes the test, all the GPT-3 models represent fine-tuned models, and GPT-3.5 are prepared models. The tasks of tests 1 and 2 are highly simplified, so the results can be quantified and evaluated by confusion matrices. Hence the performance of tests 1 and 2 are presented in Figure 7 and Figure 8. On the contrary, the tasks of tests 3 and 4 are highly integrated. Moreover, each prompt is required to process multi-tasks. Therefore, tests 3 and 4 cannot be simply identified as classification tasks.

Table 2. Results of the tests

	GPT-3.5-turbo	GPT-3.5-text-davinci-003	GPT-3.5-text-curie-001	GPT-3.5-text-babbge-001	GPT-3.5-text-ada-001	GPT-3-Curie	GPT-3-Davinci
Test 1	0	1	1	1	1	1	1
Test 2	1	1	1	1	1	1	0
Test 3	1	1	0	0	0	0	0
Test 4	1	1	0	0	0	0	0
Test 1: Prompt structure: organized, together; Instructions: required; examples: 0							
Test 2: Prompt structure: organized, together; Instructions: required; examples: 3.							
Test 3: Prompt structure: natural, separate; Instructions: required; Examples: 2.							
Test 4: Prompt structure: natural, together; Instructions: required; Examples: 3.							

According to Table 2, several essential results can be proved:

- The GPT-3 models (Curie and Davinci) could provide the same level of capabilities as the GPT-3.5 models when the GPT-3 models are fine-tuned for the specific tasks. However, the performance of the GPT-3 models is highly dependent on prompt engineering.
- In the general case, the most capable GPT-3.5 models (GPT-3.5-turbo, GPT-3.5-text-davinci-003) consistently provide the best performance among GPT models, which indicates that the inherent improvements in GPT-3.5 models may be more impactful than the fine-tuning models.
- The GPT-3.5 models show capabilities of generalization, with some models (i.e. GPT-3.5-text-davinci-003) performing better than GPT-3 models in most tests. This suggests that the fine-tuning process applied to the GPT-3 models may not be sufficient to outperform all GPT-3.5 models in building design compliance checking scenarios.

4.4.2 Quantified results

In tests 1 and 2, as the tasks are simplified to multiclassification scenarios, the performance of the models can be quantified and visualized. Figure 7 and Figure 8 present the confusion matrices of 6 models' performance in Test 1 and 2, the matrix of GPT-3.5-turbo can't be generated as the OpenAI doesn't provide GPT-3.5-turbo API reference.

As shown in Figure 7, the models with the best performance are GPT-3.5-text-davinci-003, GPT-3-curie, and GPT-3-davinci, which provide acceptable accuracy, precision, recall and F1 score in Test 1. Other models (text-curie-001, text-babbage-001, text-ada-001) present low accuracy and a high ratio of task failures which reveal these models can't implement the tasks of compliance checking.

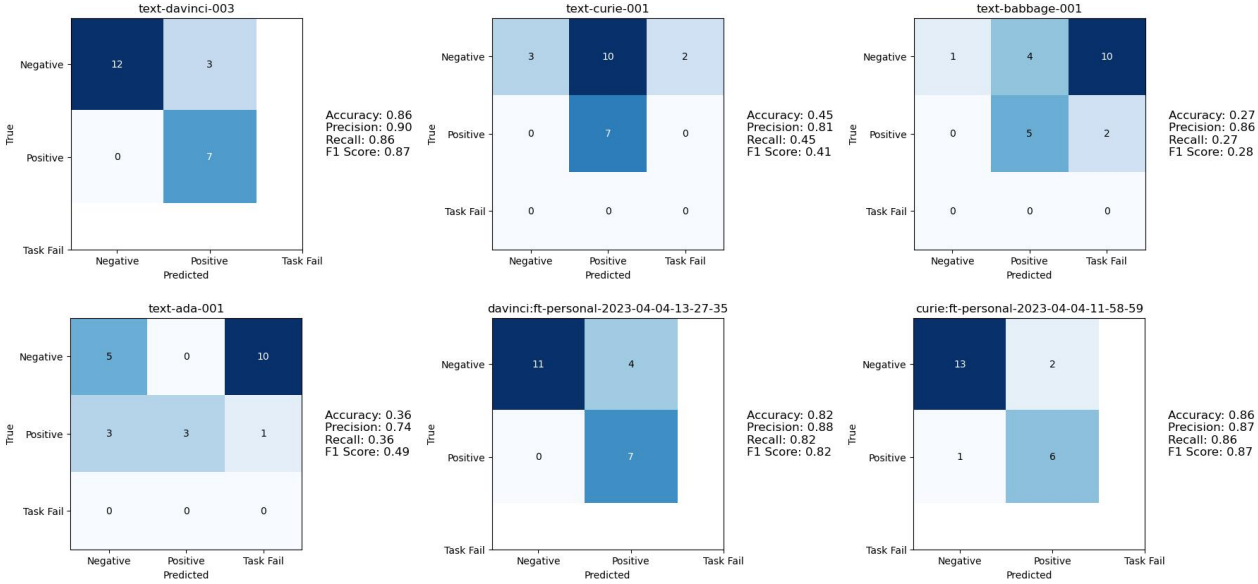

Figure 7. Confusion matrices of Test 1

In Figure 8, the davinci series model performance (GPT-3.5-text-davinci-003 and fine-tuned GPT-3-davinci) deteriorated compared with Test 1. On the contrary, other models' performance, including fine-tuned GPT-3-curie, GPT-3.5-text-curie-001, GPT-3.5-text-babbage-001, and GPT-3.5-ada-001 are improved, which proves most of the GPT-based models have a certain degree of generalization ability when the prompts structures are similar.

4.4.3 Complex analysis

Prompts of tests 3 and 4 are more challenging as LLMs are required to analyse documents which have similar structures to project documents, most of the models failed in these tests, only GPT-3.5-turbo and GPT-3.5-text-davinci-003 implemented the tasks, and the tests are not designed to be multiclassification tasks. In that case, the evaluations of the model's performance are only based on the generated completions.

In test 3, only GPT-3.5-turbo and GPT-3.5-text-davinci-003 models implement the tasks which require models to process 3 building design specifications at once, in addition, the models need to retrieve and match the regulation clauses corresponding to the design specification.

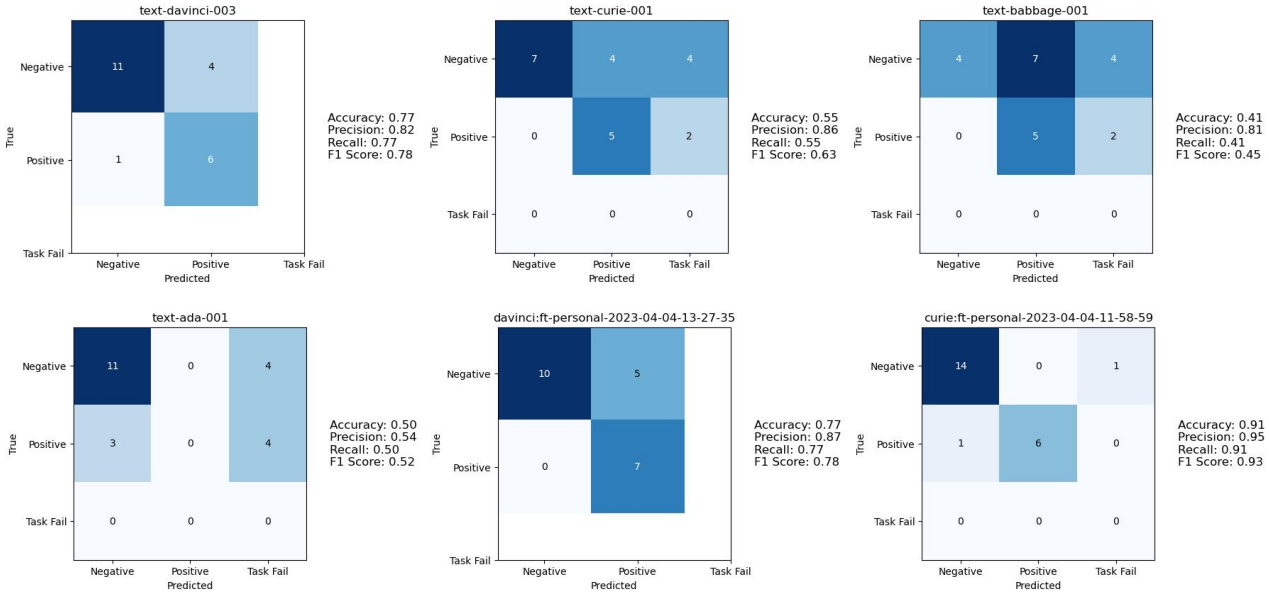

Figure 8. Confusion matrices of Test 2.

Based on test 3, test 4 requires models to generate completions in specific forms and give their explanations, moreover, unnecessary and pointless terms are added to the prompt, and the building design specification clause no longer corresponds to a single regulation term. These tasks test the comprehensive capability and robustness of the model. Only GPT-3.5-turbo and GPT-3.5-text-davinci-003 implement the tasks, though other models don't understand the task or generate unsatisfied completions.

5. Conclusion

Based on the previous results, we can draw the following conclusions:

a. The latest GPT-3.5 models, GPT-3.5-text-davinci-003, GPT-3.5-turbo provide the highest performance in state-of-art LLMs according to their accuracy, precision, recall, F1 scores, generalization ability, and robustness. GPT-3.5-text-davinci-003 is the most recommended model for adapting most types of downstream applications of complex text analysis and having strong generalization ability and robustness.

b. Fine-tuning process can largely improve models' performance. Though there is a significant capability gap between GPT-3 and GPT-3.5 basic models, the fine-tuning process can boost GPT-3 models' capability to the same level as the latest GPT-3.5 models.

c. Prompt engineering can largely determine whether LLMs can implement the target scenarios, i.e. breaking down tasks into several simple prompts can make LLMs implement the tasks easier, though more computing expanding would be costed; large, complicated prompts are easier to be generated from people, and integrated multi-tasks can be processed at the same time as long as the LLMs have enough capabilities to process them.

There are still many limitations to GPT-based models to be addressed:

- The GPT-3.5 models are not allowed to be boosted by fine-tuning yet, which means the freedom of customizing the tasks is limited, i.e. target regulations cannot be integrated

into the models by fine-tuning, they can only be learned in test prompts, which means the tasks need to be segmented into several portions;

- The model's capacity is limited to 4000 tokens (words for prompts and completions in total). In many scenarios, the total words of regulations are more than tens of thousands. Hence there would be an issue because the models cannot remember the content of several segmented but related tasks, which means the model process the segmented tasks independently and doesn't consider the correlations among several tasks.
- The computing capacity for large batch processing is restricted. During the tests in this research, the calling for multi-models' API is sometimes not allowed, it would be the issue of high frequencies of models occupying, which means the processing of a large number of documents at once is not allowed.
- The current models are "single modal ": the models only accept plain text processing for both learning and generating. The current models cannot understand the information in higher dimensions which are frequently applied in projects including sounds, images, information models etc.

Through the research, several research opportunities are proposed:

- The performance of the fine-tuned models can be further improved by modifying fine-tuning prompts dataset. There are about 160+ samples involved in the fine-tuning process, which makes the fine-tuned model GPT-3-curie achieve 91% accuracy as the best record, This would be an ideal accuracy in LLMs' performance, though the improvement of the fine-tuning dataset can boost the ability of generalization and robustness for target scenarios. The fine-tuning process can be developed by iteration, and a chain of ACC results can be generated by LLMs, not only "yes" or "not" results, the LLMs can provide explanations of the checking process and propose practical suggestions to improve the target documents through fine-tuning iterations.
- The GPT-4 model has been published. According to the demo video realized by OpenAI, GPT-4 is a multimodal LLM which can learn from images (even a rough manuscript). It would be a large development to the downstream applications of ACC in relating more drawings from the project documents and seeking for more explanations for the checking results.

References

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2018). Understanding of a convolutional neural network. Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017,

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013). Towards Automated Compliance Checking in the Construction Industry. In H. Decker, L. Lhotská, S. Link, J. Basl, & A. M. Tjoa, *Database and Expert Systems Applications* Berlin, Heidelberg.

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.

Chen, Q., & Wu, R. (2017). CNN is all you need. *arXiv preprint arXiv:1712.09662*.

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*.

Dimyadi, J., & Amor, R. (2013). Automated building code compliance checking–where is it at. *Proceedings of CIB WBC*, *6*(1).

García de Soto, B., Agustí-Juan, I., Joss, S., & Hunhevicz, J. (2022). Implications of Construction 4.0 to the workforce and organizational structures [Article]. *International Journal of Construction Management*, *22*(2), 205-217. https://doi.org/10.1080/15623599.2019.1616414

Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27,

Haitao, W., Jie, H., Xiaohong, Z., & Shufen, L. (2020). A short text classification method based on n-gram and cnn [Article]. *Chinese Journal of Electronics*, *29*(2), 248-254. https://doi.org/10.1049/cje.2020.01.001

Hassan, F. U., & Le, T. (2023). Ontology-Based Approach to Risk-Factor Identification to Support the Management of Provisions in Bridge Design [Article]. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *15*(1), Article 04522031. https://doi.org/10.1061/(ASCE)LA.1943-4170.0000574

Health, D. o. (2015). *Health Technical Memorandum 05-02: Firecode, fire safety in the design of healthcare premises*. Department of Health Retrieved from https://www.england.nhs.uk/wp-content/uploads/2021/05/HTM_05-02_2015.pdf

Indhraom Prabha, M., & Umarani Srikanth, G. (2019). Survey of Sentiment Analysis Using Deep Learning Techniques. Proceedings of 1st International Conference on Innovations in Information and Communication Technology, ICIICT 2019,

Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

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*.

Manzoor, B., Othman, I., & Pomares, J. C. (2021). Digital Technologies in the Architecture, Engineering and Construction (AEC) Industry—A Bibliometric—Qualitative Literature Review of Research Activities. *International Journal of Environmental Research and Public Health*, *18*(11), 6135. https://www.mdpi.com/1660-4601/18/11/6135

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., & Ray, A. (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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.

Ren, R., Zhang, J., Chen, Y., & Dib, H. N. (2022). A BIM information processing framework to facilitate enriched BIM applications. Construction Research Congress 2022,

Saravia, E. (2022). Prompt Engineering Guide. https://github.com/dair-ai/Prompt-Engineering-Guide

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin,

I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.
Wang, K., Guo, F., Zhang, C., Hao, J., & Schaefer, D. (2022). Digital Technology in Architecture, Engineering, and Construction (AEC) Industry: Research Trends and Practical Status toward Construction 4.0. Construction Research Congress 2022,
Zhu, Y., & Augenbroe, G. (2006). A conceptual model for supporting the integration of inter-organizational information processes of AEC projects. *Automation in Construction*, *15*(2), 200-211. https://doi.org/https://doi.org/10.1016/j.autcon.2005.05.003