Domain-specific Large Language Model Pretraining for Construction Specification Review

Shuyi Wang[1] and Jinwoo Kim[2]

[1]School of Civil and Environmental Engineering, Nanyang Technological University, Singapore
[2]Department of Civil and Environmental Engineering, Hanyang University, Seoul, Republic of Korea
shuyi003@e.ntu.edu.sg, jinwookim@hanyang.ac.kr

Abstract –
With advancements in large language models (LLMs), there has been growing attention to automatic construction specification reviews, enabling faster and more standardized assessments. However, general-purpose LLMs' effectiveness has been bounded when applied to construction contexts, because they lack necessary domain-specific vocabulary and knowledge. Therefore, we propose a domain-specialization LLM pretraining method to leverage construction terms and knowledge and quantitatively evaluate its efficacy in two specification review tasks. To this end, we generate two construction-specific corpora, i.e., close-domain and in-domain, and investigate their hybrid effects on LLM performance at varying scales. Results reveal that construction-specific contextualization led to a significant improvement, with a 6.9% increase in accuracy for extractive question answering and an 8.0% enhancement in information retrieval accuracy. We also observe that augmenting in-domain corpus with close-domain data, can achieve performance comparable to models trained with extremely large in-domain data. These will facilitate the adoption of LLMs in construction and support specification reviews.

Keywords –
Large language model; Domain-specific; Pretraining; Construction specification

1 Introduction

Construction specification review is a critical process in the construction industry, focused on ensuring that project specifications align with required standards, project goals, and regulatory compliance [1]. Specifications are detailed documents that outline project requirements, including material quality, design standards, construction methods, and safety protocols. However, manual review of such extensive documents is time-consuming and resource-intensive, leading researchers to explore automated methods using natural language processing (NLP) technologies [2,3]. Previous studies have primarily focused on two types of NLP tasks in specification reviews: (i) Contextual understanding of provisions: Its application involves interpreting and classifying regulatory provisions based on their semantic contents, such as identifying risky contractual terms [4] or categorizing provisions into classes such as technical requirements, financial regulations, and material standards [5]. This is also foundational for tasks like extractive question answering (QA) from provisions, which requires precise extraction of textual span according to user intent; and (ii) Information retrieval for specific details: Its application focuses on recognizing and extracting critical entities including subjects, requirements, actions, and standards through rule-based NLP algorithms and deep learning techniques [2,6].

Recently, large language models (LLMs), such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), have become prominent in NLP applications due to their ability to capture rich contextual information [7]. For instance, Ahmadi et al. [8] leveraged existing LLMs like Gemini and ChatGPT to classify construction accident reports into categories including injury cause, root cause, affected body part, severity, and accident type, demonstrating the promising potential of general-purpose LLMs. Although LLMs perform well in everyday language tasks, adapting them to a specific domain, particularly construction documents with many technical terminologies, presents significant challenges. In particular, LLMs pretrained on generic corpora often struggle with domain-specific vocabulary, contextual nuances, and technical language characteristic [9]. Consequently, these models lack the precise understanding required for accurately interpreting construction specifications, demonstrating the need for target domain adaptation.

Some researchers have recognized this challenge and addressed it by employing domain-adapted LLMs, such as BERT variants pretrained on domain-contextualized corpora. In this regard, two types of corpora have

typically been used: (*i*) in-domain corpus, which is narrowly focused on a specific target domain (i.e., a particular type of documents) and incorporate specialized language and contextual terminology; and (*ii*) close-domain corpus, which encompasses a wide spectrum of construction documents, capturing diverse linguistic patterns and terminology relevant to the field. For instance, Zheng et al. utilized both close-domain and in-domain Chinese corpora and validated their individual benefits as well as the combined effects of mixing two corpora on NLP-based construction document analysis. Zhong and Goodfellow [11] also introduced pretrained language models specifically designed for construction management systems, by leveraging in-domain corpora with and without sentence concentration. While these studies indicate that domain-specific pretraining enhances LLM performance in NLP tasks, several knowledge gaps remain to be addressed. First, there is limited quantitative analysis on the effects of domain-specific knowledge on LLM performance in construction specification reviews. Although untrainable models like ChatGPT demonstrate strong capabilities, a comprehensive evaluation of domain-specific adaptation on small-scale trainable models can inform their effective utilizations in construction. This trial will also help address the scarcity of labeled domain-specific data that limits supervised methods, as pretraining on unlabeled domain data offers a promising approach to reduce dependence on extensive labeled datasets [12]. Second, although previous studies theoretically identify the advantages of incorporating domain knowledge including technical vocabulary and language patterns into LLMs, few studies have investigated how varying quantities of close-domain and in-domain corpora impact model's performance. Specifically, it remains unclear how much data can motivate the model's capability in contextual understanding. Moreover, emphasis has largely been placed on in-domain corpus due to its direct relevance to target documents. However, when the available in-domain corpus is limited, its impact on model adaptation becomes marginal, making it necessary to consider supplementary close-domain data. However, the hybrid effects of combining close-domain and in-domain corpora are rarely examined, leaving unanswered questions about the optimal quantity of these corpora for efficient domain adaptation in LLM pretraining.

Therefore, we propose a LLM pretraining method to combine close-domain and in-domain corpora and specialize LLMs to the construction field. We then fine-tune and validate LLMs in two construction specification review tasks, extractive QA and information retrieval. Particularly, we evaluate the effectiveness of domain corpora on LLM performance and investigate their impacts at varying sizes. Our findings will facilitate the domain adaptation of LLMs and offer insights on optimizing training efficacy with varying domain corpora sizes.

2 Research Method

This study followed a three-stage method comprising corpus generation, LLM pretraining, and task-specific finetuning (Figure 1). First, we collected various types of publicly accessible construction documents and generated close-domain and in-domain corpora. Subsequently, we utilized masked language modeling (MLM) for LLM pretraining. Lastly, the pretrained LLMs were fine-tuned for two construction specification review tasks: extractive QA for provisions and information retrieval from provisions. More details are provided in the following subsections.

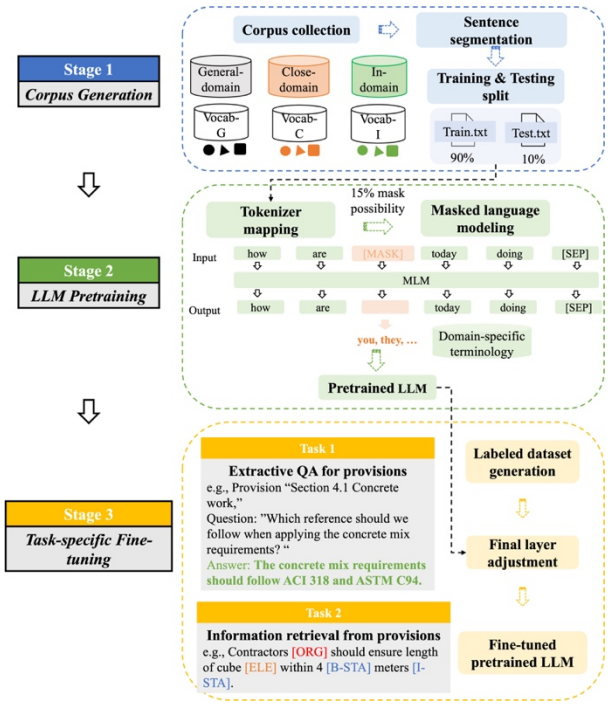

Figure 1. Overall pipeline for pretraining a construction-specific large language model

2.1 Stage 1: Corpus Generation

In Stage 1, we generated comprehensive corpora encompassing various types of documents in the construction domain, categorized into two distinct types: (*i*) Close-domain documents, which include a range of construction-related documents such as contracts, regulatory laws, and management guidelines; and (*ii*) In-domain documents, which are specific to a single document type that closely aligns with a particularly type of document (i.e., construction specifications in this study). To construct these corpora, we initially collected construction documents publicized in two different

formats. The first format consisted of Portable Document Format files, which were converted into machine-readable texts using Optical Character Recognition tools. The second format included web-based documents, for which we employed web scraping techniques to extract and consolidate textual information. Ultimately, we gathered 752 close-domain documents comprising contracts, law regulations, and management guidelines. Additionally, we obtained 56 in-domain construction specification documents from four countries, United States, United Kingdom, Canada, and Australia, covering a total of 56 distinct states and regions. Following conversions of texts for these documents, we adopted data cleaning techniques including removing extraneous characters and irrelevant formatting. We then applied text preprocessing techniques to both corpora, including sentence segmentation and training-testing splitting. Sentence segmentation enables the model to process structured linguistic units, which is critical for enhancing model understanding for complex contents. Detailed statistical information of two corpora is illustrated in Table 1. Finally, we allocated 90% of each corpus for training and 10% for testing [13]. This split ensures that the model is evaluated on a representative portion of the data while maintaining sufficient training data to capture domain-specific patterns and vocabulary.

Table 1. Statistical information of domain corpora

	Numbers of sentences	Numbers of tokens
Close-domain	1,028,086	23.6 million
In-domain	1,097,925	22.4 million

2.2 Stage 2: LLM Pretraining

After generating close-domain and in-domain corpora, we proceeded to the pretraining phase. Each sentence was initially tokenized to its numeric values that can be processed by computer programming language [10]. We then utilized MLM by randomly masking a certain percentage of input tokens within sentences, requiring the model to predict these obscured tokens. In this process, the final hidden vectors corresponding to the masked tokens were passed through an output softmax layer over the vocabulary. According to BERT implementation [13], 15% of all WordPiece tokens were randomly selected and replaced with the [MASK] token in each sequence, prompting the model to predict these tokens by learning contextual representations from neighboring words. This random selection process was managed by our data collator, which ensures 15% of token positions are designated for prediction. The cross-entropy loss function was finally computed by the average prediction error over all masked tokens in a sequence L following Eq.1 [14]:

$$Loss = -\frac{1}{|M|}\sum_{i \in M} logP(x_i|h_i^L) \quad (1).$$

where $|M|$ represents the total number of masked tokens. For each masked position i, x_i represents the original token and h_i^L is the corresponding output representation from the final layer of the model.

Notably, MLM serves as a self-supervised pretraining process, eliminating demands for additional manual labeling. Upon completion of pretraining on construction-specific corpora, this pretrained LLM acquires contextual domain knowledge, enhancing its performance in following downstream NLP tasks.

2.3 Stage 3: Task-specific Fine-tuning

Building on the pretrained LLM developed in Stage 2, we fine-tuned the final output layer by applying task-appropriate loss functions tailored for construction specification review tasks. We particularly focused on two primary tasks: extractive QA from provisions and information retrieval. With respect to the first task, we concentrated on the model's ability to identify and extract the precise span of text within a provision that answers a given question. To achieve this, we adapted the final output layer of the pretrained encoder by adding two classification heads $[Answer_s, Answer_e]$ to predict the start and end positions of the answer span within the input token sequence. During pretraining, the input consisted of a concatenated $[Provision, Question]$ pair, and the model was supervised using a span-based objective function—specifically, the cross-entropy losses for the start and end positions. This formulation enabled the model to learn fine-grained semantic alignment between the question and the contextual tokens of the provision, facilitating accurate answer localization. Meanwhile, information retrieval task refers to extract core elements from extensive documents based on their important semantic contents, which is a classical classification problem [6]. For this purpose, we appended a new classification layer $[CLS]$ to assign class labels (e.g., organization, action, element, standard, and reference) to each token based on a labeled information retrieval dataset, enabling the model to accurately categorize information of construction specifications.

3 Experiments and Results

As illustrated in Figure 2, we conducted two experiments to investigate the effects of domain-specific corpora on LLM performances (Experiment #1); and to analyze the hybrid effects of close-domain and in-domain corpora at varying sizes (Experiment #2). In both experiments, we adopted BERT as our foundational LLM due to its well-established capability in contextual understanding, which is particularly suitable for

specification review. Furthermore, it supports computationally efficient pretraining compared with larger-scale open-source LLMs, assisting practical experimentation [9]. It was then fine-tuned for two downstream tasks, extractive QA for provisions and retrieving information from provisions.

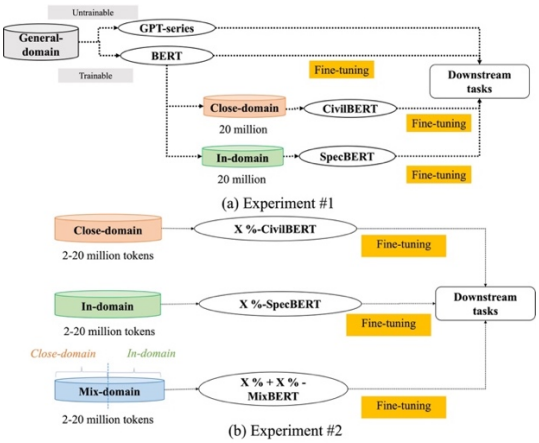

Figure 2. Experimental setups

Prior to task-specific fine-tuning, we conducted domain-adaptive pretraining to better align the LM with the semantic and syntactic characteristics of construction documents. The pretraining was carried out with the following parameters informed by [13]: a batch size of 32, a learning rate of 5×10^{-5}, a maximum sequence length of 512, and a total of 5,000 training steps. All pretraining was performed on NVIDIA A100 GPU. The evolution of both training and testing losses is demonstrated in Figure 3. It was observed that both losses converged and stabilized at around 5,000 training steps, indicating that the model reached convergence. To further validate the success and robustness of pretraining, we evaluated the model on a masked word prediction task using sentences from construction specifications. In this setting, 15% of tokens were randomly masked, and the model was tasked with recovering the original tokens. The masked token prediction accuracy reached 93.2%, demonstrating the model's strong capacity to learn and generalize over the domain terminology.

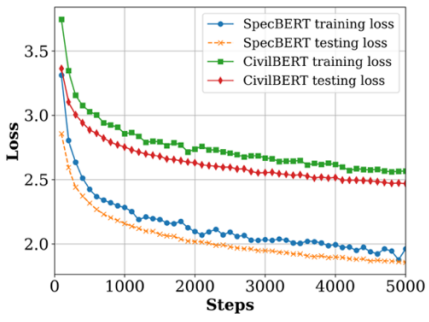

Figure 3. Losses for domain-specific pretraining

With respect to fine-tuning for experiment, we initially built fine-tuning task-specific training and test datasets as follows. To prepare the extractive QA dataset, we initially employed GPT-4o with prompt engineering to produce initial QA pairs from 201 construction provisions, as introduced in [15]. The generation process simulated real-world domain experts engaged in reviewing construction specifications, covering five question types: responsibility identification, procedure specification, numerical constraint, material standard, and regulatory references [6]. These automatically generated pairs then underwent a manual structural and semantic verification. Structurally, we ensured each pair complied with the Stanford Question Answering Dataset format—answers had to be explicitly found in the provision, with correct span indices [16]. Semantically, we first removed redundant questions that queried the same information. Second, inaccurate or contextually irrelevant questions were corrected or excluded. Third, to increase reasoning complexity, a subset of questions was reformulated. For instance, a basic question like "What curing compound is required for the concrete pouring?" was rewritten as "Which curing compound specified for concrete pouring should also satisfy water retention standard?". After this manual refinement, a total of 855 high-quality extractive QA pairs were obtained and subsequently reviewed by domain experts to ensure its quality. This dataset was divided into a training dataset (684 pairs) and testing dataset (171 pairs) with an 8:2 ratio. Using the dataset, we trained and tested models with two quantitative metrics: F1-score and Sentence-BERT (S-BERT). Specifically, F1-score provides a strict measure of token-level correspondence by balancing precision and recall, thus assessing how closely the predicted answer matches the ground truth. However, such span-based metrics penalize semantically correct answers but differ in wording. To this end, S-BERT is employed to evaluate semantic overlap by computing the cosine similarity between sentence-level embeddings of the predicted and reference answers. Collectively, these two metrics reflect both the surface-level accuracy and semantic relevance.

For the other fine-tuning task, retrieving information from provisions, a total of 1,250 provisions were extracted from the identical specifications used in the former task. These provisions were then categorized into five classes according to [6] to extract specification review information: Organization, Action, Element, Standard, and Reference. Each token was manually labeled as one entity using the open-source annotation tool Label Studio, with contiguous tokens assigned consistent labels and explicit boundaries defined to prevent overlap between entity types. In total, 15,969 tokens were annotated with corresponding semantic entity labels. This dataset was then divided into a training

dataset (1,000 provisions) and testing dataset (250 provisions) with an 8:2 ratio. To assess the model's performance, two evaluation metrics, F1-score and accuracy, were utilized. Specifically, F1-score integrates both precision and recall and requires a predicted entity match the gold standard in both its semantic label and boundary to be counted as correct. In contrast, accuracy measures performance at the token level by calculating the proportion of tokens assigned the correct label, including non-entity ("O") tokens. More detailed settings and results of each experiment are provided in the following subsections.

3.1 Experiment #1: Effects of Domain-specific Corpora on LLM

In Experiment #1, we compared the performance of four types of LLMs in construction specification review tasks: (i) an untrainable LLM like GPT-4o; (ii) BERT, which is a general-purpose LLM pretrained on a broad corpus; (iii) CivilBERT, a model pretrained on a close-domain corpus of 20 million tokens; and (iv) SpecBERT, a model pretrained on an in-domain corpus of 20 million tokens. Baseline models were determined as untrainable and pretrained LLMs from general corpus (i and ii).

3.1.1 Downstream task #1 results: Extractive QA for provisions

As illustrated in Table 2, general-purpose LLMs, such as GPT-4o, exhibited relatively lower performance. Specifically, GPT-4o achieved an F1-score of 74.0% and an S-BERT of 79.1% on the extractive QA. The BERT model, pretrained on a broad general corpus and fine-tuned on the extractive QA dataset, showed a modest improvement of 6.4% in F1-score and 4.0% in S-BERT, reaching final scores of 78.8% and 82.3% respectively. It was noteworthy that LLMs pretrained with domain-specific data, however, displayed significantly enhanced performance. CivilBERT attained an F1-score of 80.1% and an S-BERT of 86.4%, surpassing BERT by 1.4% in F1-score and 4.1% in S-BERT. Moreover, SpecBERT delivered the highest performance, with an F1-score of 82.4% and S-BERT of 89.2%, representing increases of 3.7% and 6.9% over BERT in two metrics. Overall, these results indicated that with domain-specific pretraining, smaller-scale LMs could surpass out-of-the-box commercial LLMs on specialized extractive QA tasks.

Table 2. Comparisons of LLM performances on extractive QA dataset

	F1-score	S-BERT
GPT-4o	74.0%	79.1%
BERT	78.7%	82.3%
CivilBERT	80.1%	86.4%
SpecBERT	**82.4%**	**89.2%**

Figure 4 presents a representative example of qualitative behaviors of baselines and domain-specific LLMs. Specifically, GPT-4o tended to prioritize completeness in answering the question by extracting all information related to the question and adding additional descriptions inferred from the contents. While this resulted in comprehensive responses, it often introduced unnecessary elaboration beyond the exact answer span. In contrast, BERT fine-tuned on the dataset adhered strictly to the question and focus on retrieving spans that appear directly relevant. However, due to the lack of pre-existing semantic understanding of the entire provision, BERT often extracted multiple loosely related segments. CivilBERT and SpecBERT, on the other hand, demonstrated stronger contextual comprehension. Both models exhibited a better grasp of how to localize and extract answer spans based on the semantic alignment between the question and the provision. This reflected their enhanced capacity for domain-aware understanding and more accurate extractive reasoning.

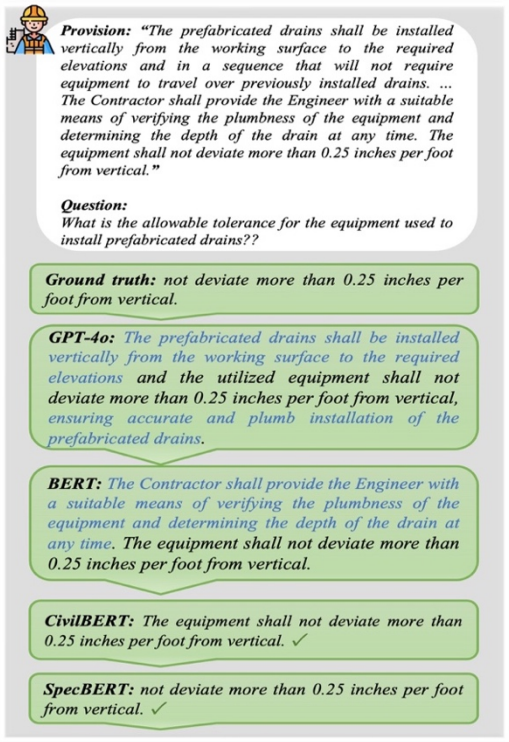

Figure 4. Qualitative behaviors of baselines and domain-specific LLMs for extractive QA pair

3.1.2 Downstream task #2 results: Information retrieval from provisions

Table 3 presented comparative results across three corpus types for the information retrieval dataset. General-purpose LLMs demonstrated relatively lower performance, with GPT-4o achieved F1-score at 53.9% and accuracy at 54.6%. BERT displayed substantial

improvement, with an F1-score of 71.7% and accuracy of 84.1%. Domain-specific pretraining further enhanced performance. CivilBERT outperformed BERT with an F1-score of 76.3% and accuracy of 87.7%, representing increases of 6.4% and 4.3% in two metrics, respectively. The best performance was observed in SpecBERT, which achieved an F1-score of 77.4% and accuracy of 88.9%, surpassing BERT by 8.0% in F1-score and 5.7% in accuracy compared to BERT.

Table 3. Comparisons of LLM performances on information retrieval dataset

	F1-score	Accuracy
GPT-4o	53.9%	54.6%
BERT	71.7%	84.1%
CivilBERT	76.3%	87.7%
SpecBERT	**77.4%**	**88.9%**

As illustrated in Figure 5, GPT-4o demonstrated limited capability to accurately classify domain-specific terms due to the lack of prior construction-specific knowledge in its general-purpose pretraining. To be specific, it failed to recognize specialized terminology such as "geotechnical fill materials," instead generalizing to broader terms like "materials." Additionally, its interpretation of standards and references tended to be ambiguous, frequently resulting in overlapping or misclassified outputs. In contrast, the fine-tuned BERT model showed improved performance, demonstrating the capability to distinguish between standards and references, while also preserving the integrity of domain-specific terminology. CivilBERT further enhanced this performance, though it struggled with certain terms, for instance, simplifying "resident engineer" to "engineer," and missed critical references, especially those pertaining to project-specific or organization-defined specifications beyond national standards. Notably, SpecBERT achieved the best overall performance. Benefiting from domain-specific pretraining, it was capable of extracting a more comprehensive set of relevant terms and correctly categorizing them semantically.

Overall, these results highlighted that domain-specific pretraining, particularly with in-domain corpora (e.g., SpecBERT), substantially enhanced performance of LLMs in both contextual comprehension and information retrieval. By integrating construction-specific technical terminologies, the model's ability to capture contextual contents and extract semantic elements was markedly strengthened, addressing unique linguistic complexities of the construction domain. This domain-aware capability proves especially valuable across a range of practical construction management applications [8,11]. Specifically, it facilitates information retrieval for decision support, allowing site engineers and project managers to quickly locate technical requirements—such as materials, construction methods, and tolerances—during on-site execution. Through aligning the model's linguistic capacity with the operational realities of construction workflows, domain-specific pretraining contributes to a more intelligent, efficient, and trustworthy document processing pipeline**,** supporting both high-level regulatory reviews and real-time field-level implementation.

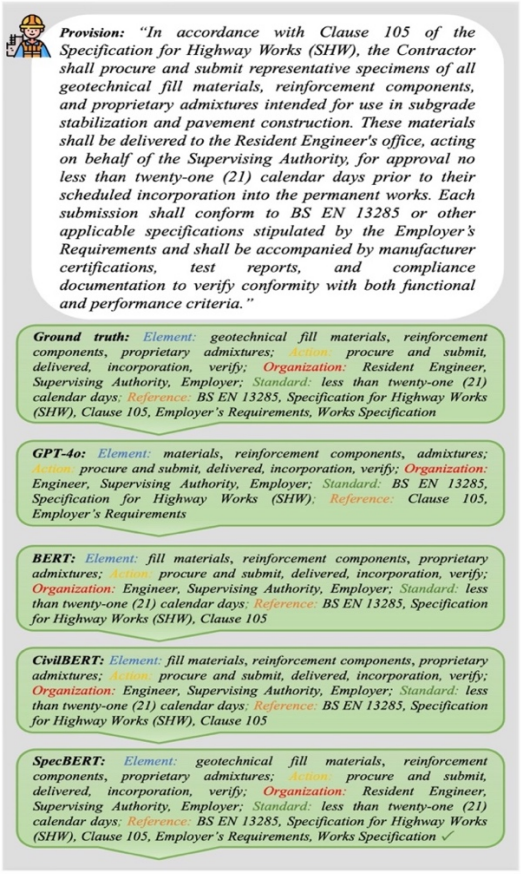

Figure 5. Qualitative behaviors of baselines and domain-specific LLMs for information retrieval

3.2 Experiment #2: Hybrid Effects of Close-domain and In-domain Corpora on LLM

Experiment #2 examined the effects of varying corpus quantities by evaluating different proportions $x\%$ ($x \in \{10, 20, 30, 40, 50, 100\}$) of corpus data from single domains (close-domain or in-domain) and mixed domains in finetuning tasks. A maximum of 20 million tokens from each corpus was defined as the full-scale setting. Accordingly, 2 million tokens of in-domain data produced a 10%-SpecBERT, while mixture of 2 million tokens from each domain (total 4 million tokens) yielded a 10%+10%-MixBERT. Baseline models were established as LLMs pretrained on single-domain data with the same corpus size as the mixed-domain LLMs.

3.2.1 Downstream task #1 results: Extractive QA for provisions

Figure 6 demonstrated model performance varying with corpora quantity in extractive QA for provisions. It was found that incorporating 10% close-domain data (2 million) or 10% in-domain data (2 million) led to a slight increase in S-BERT to 82.6% and 83.1%, respectively, surpassing the baseline (82.3%). As the corpus size ranged from 2 million to 20 million tokens, models pretrained exclusively on in-domain data improved from 83.1% to 89.2% while those pretrained on a mixture of close-domain and in-domain data increased from 83.8% to 88.1%. In contrast, models pretrained totally in close-domain corpus exhibited the lowest performance. These findings indicated the substantial advantages of maximizing in-domain data for contextual comprehension. Meanwhile, when in-domain data was limited, augmenting the in-domain corpus with close-domain data during pretraining really enhanced model performance. For instance, a model pretrained on mixture of 4 million tokens of in-domain data and 4 million tokens of close-domain data attained a S-BERT of 85.9%, outperforming the 84.7% S-BERT of a model pretrained exclusively on 4 million tokens of in-domain data.

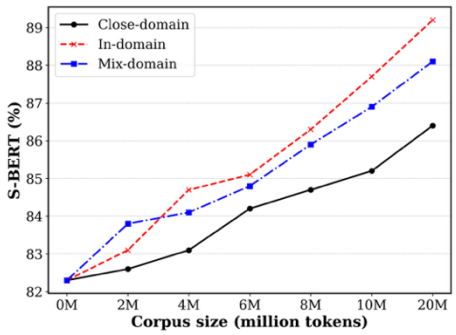

Figure 6. Model performance varying with corpora quantity in extractive QA

3.2.2 Downstream task #2 results: Information retrieval from provisions

Figure 7 illustrated how model performance varies with corpus size across three domains in information retrieval. Initial results showed that with a limited corpus of 2 million tokens (10%), models pretrained on close-domain and in-domain data achieved F1-scores of 71.1% and 71.4%, respectively, both slightly below the F1-score of 71.7% for the model without domain corpus. However, a hybrid approach combining 5% close-domain and 5% in-domain data yielded a notable improvement with an F1-score of 73.3%, a 1.9% increase over model pretrained from 10% in-domain data. This improvement suggested that the hybrid strategy effectively mitigated the overfitting risk associated with small amounts of either close-domain or in-domain data alone. As corpus size increased from 4 million to 20 million tokens, models pretrained on hybrid corpora exhibited a consistent upward trend in F1-scores, ranging from 73.5% to 76.8%. These performances were comparable to those of models pretrained exclusively on in-domain data from 73.9% to 77.4%, much higher than models pretrained using only the close-domain corpus.

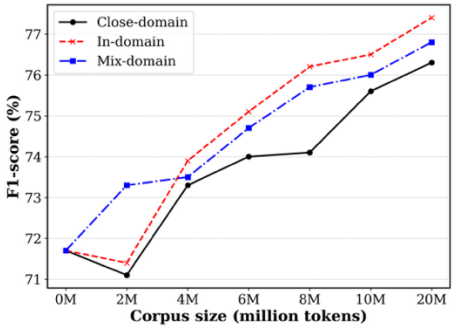

Figure 7. Model performance varying with corpora quantity in information retrieval

4 Conclusion

In this study, we proposed a three-stage LLM pretraining method to integrate domain-specific terminologies, enhancing its generalization and performances in construction specification review tasks. First, we constructed both close-domain and in-domain corpora to capture relevant construction-specific vocabulary and contexts. We then applied MLM for pretraining on these different types of corpora. Finally, we fine-tuned these pretrained models for two specification review tasks: extractive QA for provisions and information retrieval from provisions. To validate the effectiveness, we conducted two experiments. In Experiment 1, SpecBERT, our in-domain pretrained LLM, achieved the highest and at 82.4% and 89.2% in F1-score and S-BERT, surpassing the fine-tuned BERT model by 3.7% and 6.9% respectively, Additionally, SpecBERT demonstrated superior results in information retrieval from provisions, achieving a 77.4% F1-score and 88.9% accuracy, outperforming the state-of-the-art BERT model by 7.9% and 5.7%, respectively. Experiment #2 evaluated varying quantities and hybrid effects of close-domain and in-domain corpora. Results demonstrated that increasing quantity of domain corpora benefits the performance of LLMs in downstream tasks. When the size of in-domain corpus is limited, it is advisable to incorporate additional close-domain corpus data. Notably, a hybrid approach, mixture of close-domain and in-domain corpora, can yield performance levels comparable to models trained on sufficient quantities of in-domain data alone. In overall, this study

offered a practical approach to develop domain-aware pretrained LLMs adaptable to various construction specification review tasks under varying quantity.

Despite the promising potential of the work, there still exist two limitations. First, the size of the fine-tuning dataset was relatively limited. Increasing the volume and diversity of the dataset could further enhance model performance and robustness. Second, the pretraining corpus was English-centric, which may limit the model's generalizability to specifications in other languages. Future work will extend the approach to multilingual settings through cross-lingual pretraining and adaptation.

Acknowledgement

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant RS-2026-25524415).

References

[1] S. Moon, G. Lee, S. Chi, Automated system for construction specification review using natural language processing, Advanced Engineering Informatics 51 (2022) 101495. https://doi.org/10.1016/j.aei.2021.101495.

[2] J. Zhang, N.M. El-Gohary, Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking, J. Comput. Civ. Eng. 30 (2016) 04015014. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346.

[3] R. Zhang, N. El-Gohary, Natural language generation and deep learning for intelligent building codes, Adv. Eng. Inf. 52 (Apr) (2022) 101557. https://doi.org/10.1016/j.aei.2022.101557.

[4] S. Moon, S. Chi, S.-B. Im, Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (BERT), Autom. Constr. 142 (Oct) (2022) 104465. https://doi.org/10.1016/j.autcon.2022.104465.

[5] J. Zhou, Z. Ma, Named Entity Recognition for Construction Documents Based on Fine-Tuning of Large Language Models with Low-Quality Datasets, (2024). https://doi.org/10.2139/ssrn.4914418.

[6] S. Moon, G. Lee, S. Chi, H. Oh, Automated Construction Specification Review with Named Entity Recognition Using Natural Language Processing, Journal of Construction Engineering and Management 147 (2021) 04020147. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001953.

[7] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Dou, J.-R. Wen, Large Language Models for Information Retrieval: A Survey, (2024). http://arxiv.org/abs/2308.07107 (accessed July 9, 2024).

[8] E. Ahmadi, S. Muley, C. Wang, Automatic Construction Accident Report Analysis Using Large Language Models (LLMs), J. Intell. Constru 3 (2024) 9180039. https://doi.org/10.26599/JIC.2024.9180039.

[9] Y. Kim, J.-H. Kim, Y.-M. Kim, S. Song, H.J. Joo, Predicting medical specialty from text based on a domain-specific pre-trained BERT, International Journal of Medical Informatics 170 (2023) 104956. https://doi.org/10.1016/j.ijmedinf.2022.104956.

[10] Z. Zheng, X.-Z. Lu, K.-Y. Chen, Y.-C. Zhou, J.-R. Lin, Pretrained domain-specific language model for natural language processing tasks in the AEC domain, Computers in Industry 142 (2022) 103733. https://doi.org/10.1016/j.compind.2022.103733.

[11] Y. Zhong, S.D. Goodfellow, Domain-specific language models pre-trained on construction management systems corpora, Automation in Construction 160 (2024) 105316. https://doi.org/10.1016/j.autcon.2024.105316.

[12] L. Zheng, N. Guha, B.R. Anderson, P. Henderson, D.E. Ho, When does pretraining help?: assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, ACM, São Paulo Brazil, 2021: pp. 159–168. https://doi.org/10.1145/3462757.3466088.

[13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, (2019). http://arxiv.org/abs/1810.04805 (accessed April 17, 2024).

[14] J. Salazar, D. Liang, T.Q. Nguyen, K. Kirchhoff, Masked Language Model Scoring, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: pp. 2699–2712. https://doi.org/10.18653/v1/2020.acl-main.240.

[15] S. Wang, Y. Fu, J. Kim, Toward construction-specialized, small language models: The interplay of domain adaptation, model scale and data volume, Adv. Eng. Inf. 69 (Jan) (2026) 104035. https://doi.org/10.1016/j.aei.2025.104035.

[16] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ Questions for Machine Comprehension of Text, (2016). https://doi.org/10.48550/arXiv.1606.05250.