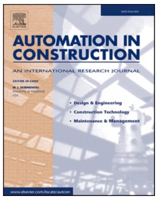
Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Review

Domain-specific language models pre-trained on construction management systems corpora

Yunshun Zhong [*], Sebastian D. Goodfellow

Dept. of Civil and Mineral Engineering, Univ. of Toronto, 35 St. George St, Toronto, ON M5S 1A4, Canada

ARTICLE INFO

Keywords:
Construction management
Domain-specific large language models
Pre-training
Natural language processing (NLP)
Transfer learning
Text classification (TC)
Named entity recognition (NER)
Corpus development

ABSTRACT

The rising demand for automated methods in the Construction Management Systems (CMS) sector highlights opportunities for the Transformer architecture, which enables pre-training Deep Learning models on large, unlabeled datasets for Natural Language Processing (NLP) tasks, outperforming traditional Recurrent Neural Network models. However, their potential in the CMS domain remains underexplored. Therefore, this research produced the first CMS domain corpora from academic papers and introduced an end-to-end pipeline for pre-training and fine-tuning domain-specific Pre-trained Language Models. Four corpora were constructed and transfer learning was employed to pre-train BERT and RoBERTa using the corpora. The best-performing models were then fine-tuned and outperformed models pre-trained on general corpora. In two key NLP tasks, text classification using an infrastructure condition prediction dataset and named entity recognition using an automatic construction control dataset, domain-specific pre-training improved F1 scores by 5.9% and 8.5%, respectively. These promising results demonstrate extended applicability beyond CMS to the Architecture, Engineering, and Construction sectors.

1. Introduction

Machine Learning (ML) and Deep Learning (DL) have been widely used in construction management Systems (CMS) for modeling structured data, whereas the application for unstructured text data analysis is still nascent. Text data, such as inspection reports, contain valuable information on infrastructure conditions but are underutilized due to their unstructured format. With over 80% of CMS data being unstructured and predominantly textual, this represents a significant barrier to effective data utilization [1]. For instance, the evaluation of building designs has traditionally been conducted manually by domain experts. This method, while reliant on expert knowledge, inherently introduces subjective biases, thus limiting the capability to efficiently and sustainably manage the vast and diverse array of building information and regulations [2]. Furthermore, the process of analyzing near-miss incidents in safety reports is notably laborious and time-consuming, posing significant challenges in terms of resource allocation and timely response [3]. These scenarios underscore the critical need for adopting data-driven methodologies that can transcend the constraints of conventional, labor-intensive practices in the field. Consequently, efforts have been directed toward applying DL-based Natural Language Processing (NLP) technologies in the CMS, including automated compliance checking (ACC) [4], asset condition prediction [5], and filtering information [6].

In general, the DL-based NLP methodologies employed in the CMS sector predominantly encompass these two tasks [7]:

- Text classification (TC) involves assigning predefined categories (or labels) to a text based on its content. Example applications in the CMS domain include hierarchical text classification for ACC [8], construction site accident classification using documents [9], and near-miss information classification from safety reports [3].
- Named entity recognition (NER) identifies and classifies named entities in a text into predefined categories. Example applications in CMS domain include IFC-regulation semantic information alignment [4], rule-based electrical and plumbing information extraction [10], and extraction of requirements from regulatory documents into computer-processable representations [11].

DL methods typically require a larger amount of data for training and have significantly more parameters than traditional machine learning models [11]. In the realm of CMS, this poses a significant challenge due

[*] Corresponding author.
E-mail addresses: yunshun.zhong@mail.utoronto.ca (Y. Zhong), sebastian.goodfellow@utoronto.ca (S.D. Goodfellow).

https://doi.org/10.1016/j.autcon.2024.105316
Received 17 September 2023; Received in revised form 14 December 2023; Accepted 1 February 2024
Available online 12 February 2024

to the scarcity of publicly available, large-scale training datasets with unified semantic labels. The scarcity of such datasets necessitates substantial manual effort for dataset preparation, making it both resource-intensive and expensive [12]. This challenge is exacerbated by the heterogeneous nature of CMS data, which includes a variety of document types and formats, further complicating the data preparation process. Moreover, evaluating and comparing the performance of various DL models within CMS is fraught with difficulties. Each model may be trained on distinct datasets with varying quality and scope, leading to inconsistencies in performance metrics and benchmarking standards. This variability hinders the development of universally applicable and robust DL solutions in the CMS domain. Recently, Large-scale Pre-trained Language Models (PLMs) such as bidirectional encoder representation from transformers (BERT) [13] and generative pre-training (GPT) series [14] have made significant strides, and become a major achievement in the field of artificial intelligence (AI) [15].

Pre-training and fine-tuning a neural network is a process that comprises an initial training phase, which is typically conducted on a large and unlabeled dataset using self-supervised learning techniques. This is followed by a fine-tuning phase in which the model obtained from the initial training phase is further optimized on a downstream task or dataset using supervised learning methods. By fine-tuning the model parameters for specific tasks, the extensive knowledge implicitly contained within them can be used to enhance the performance of a wide range of downstream tasks. It is currently the general agreement within the AI community to utilize PLMs as the foundation for downstream tasks, rather than training models from scratch [15].

Large-scale pre-trained language models are predominantly trained on general-domain corpora, which usually have a different word embedding compared to domain-specific corpora. A word embedding is a learned representation for text where words that have the same meaning have a similar representation. The quality of the word embedding usually has a significant influence on the model's accuracy. To enable DL models to learn, every word must be represented as a real-valued vector in a predefined vector space. For example, the BERT model uses WordPiece embeddings [16] with a 30,000 token vocabulary in which the word embeddings are based on general English dictionaries, as shown in Fig. 1a.

It should be noted that many words in the field of civil engineering possess distinct meanings in comparison to their usage in the general English lexicon. An illustration of this can be found in the word”moment,” which in the field of civil engineering denotes a force that induces rotation or bending in a structure, whereas, in the general English lexicon, it usually denotes a brief period. It is imperative to note that the differences in terminology may result in an inability to capture the exact meaning of terminologies in the CMS domain and can result in suboptimal model performance when applied directly to CMS-related tasks. Additionally, the issue of domain specificity in PLMs is a critical gap. The general-domain training of PLMs overlooks the specific linguistic and contextual intricacies of CMS-related texts, leading to a lack of precision and accuracy in tasks such as ACC, risk assessment, and asset condition prediction.

This gap in domain-specificity within PLMs highlights the importance of applying transfer learning using domain-specific corpora, which are carefully tailored to meet the unique linguistic demands of the CMS field [7]. Transfer learning, as defined by Weiss et al. [17], involves enhancing a model's performance in one domain by transferring knowledge from a related domain. In this context, the pretraining of PLMs on CMS domain corpora constitutes a form of transfer learning, where the model initially pre-trained on a general corpus is further pre-trained using specialized CMS domain data.

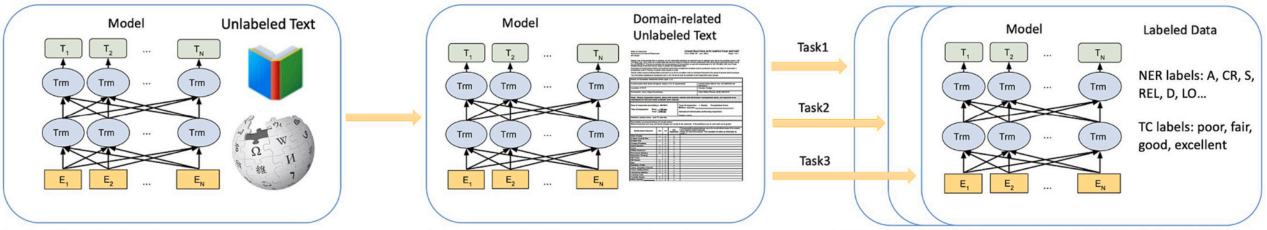
Developing the first corpora in the CMS domain and PLMs trained on domain-related corpora is important for several reasons [2,7,11,18–20]: (1) Data availability: Civil engineering datasets are typically smaller and more specialized than those used to train general-purpose models. Pre-training on domain-related corpora allows us to better leverage the available data and improve the performance of models on civil engineering tasks. (2) Domain knowledge: Pre-training on domainrelated corpora allows the model to learn the specific terminology and concepts used in civil engineering, which can improve its performance on civil engineering tasks. (3) Transfer learning: Pre-trained models can be fine-tuned on smaller, task-specific datasets, which can reduce the need for large amounts of labeled data. (4) Efficiency: Pre-training allows for faster training times and better results by utilizing knowledge already learned from the pre-training task. (5) Cost: Collecting and labeling large amounts of data for training can be expensive. Pre-training on a related corpus can reduce the need for additional labeled data and thus, reducing the cost of training models.

To bridge these gaps, the current investigation presents the development of four open-source corpora in the CMS domain and the pre-training of large language models (LLM) including BERT and RoBERTa using the developed CMS domain corpus. Additionally, an ablation study, a comprehensive evaluation to systematically analyze the impact of various elements of the model, is performed. This includes examining the effects of different pre-training techniques, hyperparameters, the choice of LLMs, and data-cleaning methodologies.

The structure of the remainder of this paper is delineated as follows. Section 2 provides a comprehensive review of the work related to this topic. The strategies for pre-training and fine-tuning PLMs on domain-specific corpus and datasets are elaborated upon in Section 3. The development of the dataset and the procedures for data cleaning and pre-processing are detailed in Section 4. Section5 not only depicts the experimental setup and results but also facilitates an analytical discussion of them. Last but not least, the advantages, contributions, conclusion, potential limitations, and prospective directions for future investigations of this research are discussed in Section 6.

2. Literature review

Early research on rule-based NLP applications in construction management has contributed to the successful extraction of information from textual data in this domain [21] [22] [23]. For example, Xu et al. developed a rule-based NLP approach to extracting domain knowledge elements (DKEs) from Chinese text documents in the domain of construction safety management [23]. However, the rules are defined with respect to their own dataset. Rule-based models usually have a hard time



(a) PTM pretrained on general corpora (b) Pretrain PTM again on domain corpora (c) Fine-tune PTM for different CMS-related NLP tasks

Fig. 1. Overall pre-training and fine-tuning procedures for our model.

generalizing to another dataset to cover variant scenarios, especially considering the heterogeneous nature of the inspection reports composed by different professional inspectors. Existing ML based applications typically rely on leveraging word frequency features [8] or syntactical features [24], which provides a certain level of automation. For example, Zhou et al. [8] developed a machine learning-based TC algorithm for classifying clauses in environmental regulatory documents based on the TC topic hierarchy. Such an approach treats each word as an atomic symbol. With such representation, one often ends up with huge sparse vectors. In addition, the relationship between any pair of words is often ignored, which restricts the model's ability to take advantage of the semantics in inspection reports. The application of context-aware DL-based NLP methods in construction management is relatively limited, but the complexity of tasks requires such a model to capture both the words and the contexts so that it can extract accurate information and achieve high accuracies on various datasets. Li et al. [25] employed a bi-directional Long Short Term Memory (LSTM) neural network architecture that is able to automatically extract information from the raw textual data in bridge inspection reports into five condition categories. The inherently sequential nature of recurrent models precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples [26]. For Recurrent models such as bi-directional LSTM, it is challenging to speed up the computation with parallelization and process long sequences such as structural inspection reports. The reason is that the decoders of LSTM only has access to the final hidden states from the encoder and it's hard to summarize long sentence in a single vector.

Transformer, a new deep learning architecture that is more parallelizable than recurrent models, has already been used in NLP tasks in the domain of construction management. Transformer is proposed by Vaswani et al. [26] in the computer science domain which relies entirely on attention mechanism [27] to calculate all dependencies between input and output. Transformer has achieved better accuracies and BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost in the year that it was published [26]. What's more important is that Transformer's ability for computation with parallelization enables the pre-training and transfer learning for NLP tasks. Pre-training means training model parameters on some tasks and then initializing the model parameters of new tasks with previous parameters. Transfer learning means taking the relevant parts of a pre-trained model and applying it to a new but similar problem. Jacob et al. [13] proposed a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks such as language inference and question answering and without substantial task-specific architecture modifications [13].

Transformer-based models are able to transfer learning from one domain to another domain. Using it for text-related analysis in construction management may enhance accuracy and make the automation of analysis more feasible. Transformer architecture will speed up computation, and help solve gradient vanish or explosion problems caused by long sequences for current machine learning and recurrent models, especially in Recurrent Neural Networks (RNNs) [28]. RNNs are a class of neural networks that are suitable for processing sequences of inputs, such as text. However, they often struggle with long-range dependencies due to the gradient issues [29]. Furthermore, pre-training usually requires less effort in building the model's architecture. Finally, Transformer-based models typically increase the accuracy of predictions compared to RNN-based models.

Zheng et al. [7] proposed the first domain corpora in Chinese and enhanced DL-based transfer learning techniques for various NLP tasks in the Architecture, Engineering, and Construction (AEC) domain. In the evaluation of all NLP tasks, it was found that the PLMs such as BERT pre-trained on domain-specific corpora demonstrated superior performance when compared to PLMs pre-trained on general corpus. The improvement for embedding-based DL models was observed in the weighted F1 score for TC at 6.4%, and in the macro F1 score for NER tasks at 5.4%. The F1-score, being the harmonic mean of precision and recall, provides a more comprehensive evaluation of a model's performance. The weighted F1-score in TC accounts for the importance of each class in proportion to its representation in the dataset, thus ensuring a balanced assessment across all categories. Conversely, the macro F1-score in NER considers each entity category equally, regardless of its frequency, thereby ensuring that rare entities are given equal importance in the evaluation. While these advancements constitute a significant step forward, it is also worth noting that some areas in this paper may benefit from further exploration and refinement. 1) A considerable part of the corpora comes from sources such as Wikipedia and other online-crawled text. Since the PLMs used were initially trained on a diverse range of online data, including Wikipedia, this scenario could potentially render their pretraining process akin to fine-tuning. Future research could consider distinct datasets to mitigate this overlap. 2) The focus on the Chinese language in the dataset limits its direct applicability to tasks in English. This constraint opens an opportunity for future work to construct English domain corpora, thus extending the applicability of these models and techniques to tasks in English.

Furthermore, it should be noted that various improvements can be made in regard to pre-training, word embedding, optimization techniques, and the selection of deep learning models, as discussed in the paper by Zheng [7]. In light of the absence of an English CMS domain dataset suitable for pretraining, this research will engage in the development of the first CMS domain corpus. The text will be extracted from academic publications for several compelling reasons. Firstly, academic papers present a novel data source. A majority of PLMs do not incorporate the text in scholarly papers during pretraining, mainly due to copyright considerations and data format challenges, especially since academic papers are commonly found in PDF format. Secondly, academic publications offer reliability. They generally undergo a rigorous process of peer review before being published, indicating that they have been critically evaluated and approved by field experts. This review process assures the reliability of the content and its contribution to the body of knowledge. Thirdly, scholarly papers ensure accuracy. The stringent editorial standards of academic publishing mean that these papers maintain a high level of accuracy in both content and language use. This precision is beneficial for NLP tasks requiring detailed semantic understanding. Lastly, the content depth of scholarly papers usually surpasses the content found in average online text data. These documents often contain thorough analyses, in-depth research, and extensive discussions on specific topics. This makes academic publications a valuable resource for numerous NLP tasks.

To address these needs, this research will be developed in four steps: (1) Developing the first corpus in the English language specific to the domain of CMS; (2) Pre-train PLMs that have been pretrained on a general corpus with domain corpus; (3) Assessing the efficacy of domain-specific PLMs in comparison to PLMs that are only pre-trained on general corpora and baseline models; and (4) Conducting a gird search of hyperparameters to further improve model performance.

3. Research methodology

This section outlines the research methodology employed to investigate the efficacy of pre-trained deep learning models on domain-specific tasks within the CMS domain. The central premise of this study is that deep learning models, initially pre-trained on a general corpus as shown in Fig. 1a, will benefit from additional pre-training on a CMS-specific domain corpus. This additional pre-training phase, depicted in Fig. 1b, is hypothesized to enhance the models' comprehension of domain-specific nuances, thereby optimizing their performance in downstream NLP tasks, a.k.a. CMS-related NLP tasks such as TC and NER, without necessitating increased manual annotation efforts. The overall procedure for pre-training and fine-tuning is shown in Fig. 1b and c. The methodology of this research is further detailed through the workflow depicted in Fig. 2. The proposed workflow consists of three parts: (1) Domain corpora development: This phase involves the development of four unique domain corpora, each subject to differing data cleaning and pre-processing techniques. Further details of this stage can be found in Section 4.1. (2) Pre-train PLMs on domain corpora - a detailed exposition of this phase is available in Section 3.1. (3) The fine-tuning of pre-trained domain models - this phase encompasses the development and assessment of several DL models of varying architectural designs for TC and NER tasks. A more extensive explanation of this step is provided in Section 3.2.

The entirety of the workflow is implemented in Python 3.9 and leverages a multitude of Python packages including Transformers, PyTorch, Selenium, Pandas, Pickle, Sklearn, Matplotlib, and Numpy, to facilitate the development of our models.

The research presented herein employs two prominent PLMs: the BERT (Bidirectional Encoder Representations from Transformers) model, as conceptualized by Devlin et al. [13], and the RoBERTa (A Robustly Optimized BERT Pretraining Approach) model, developed by Liu et al. [30]. Both models are built upon the innovative architecture of the Transformer encoder, which is illustrated in Fig. 3a. Specifically, the encoding procedure of Transformer encoder is shown in Fig. 3b, which will be discussed in detail in Section 3.1.1. The RoBERTa model represents a sophisticated evolution of the original BERT architecture, distinguished by several key enhancements. These include an extended period of training, larger batch sizes during this phase, and a broader spectrum of training data. Notably, RoBERTa omits the next sentence prediction task, a feature of its predecessor, in favor of training on extended sequence lengths. Additionally, it incorporates a dynamic approach to altering the masking pattern employed in the training data, further refining the learning process.

3.1. Pretrain PLM on domain corpus

The series of steps constituting the pre-training procedure, as diagrammatically represented in the middle section of Fig. 2, will be elucidated in sequential order in the following Section 3.1.1, 3.1.3, and 3.1.4.

3.1.1. Word embedding methods

Word embeddings serve as a method for encoding semantic information in words, where learned representations of text enable words with analogous meanings to exhibit similar representations. Essentially, this is a form of word vectorization, a technique to convert textual data into a numerical form that can be comprehended by a machine. The quality of these word embeddings frequently exerts a substantial impact on the accuracy of a model. Utilizing word embeddings necessitates that every word is represented as a real-valued vector within a predefined vector space, facilitating machine learning algorithms to comprehend and learn from the text data.

In their research, Zheng et al. [7] employed the skip-gram model with negative sampling [31] as their word embedding technique for deep learning models that rely on static word embeddings. However, this approach presents a limitation as each word can only possess a singular representation, not considering the context in which the word is used. Contrasting traditional word embeddings like Word2Vec [32] or GloVe [33], where each word or token has a single static vector representation, the transformer architecture fosters dynamic word embeddings. This implies that the representation of a word is contingent upon the context of its use in a sentence, thereby enabling a more nuanced understanding of word meanings. Consequently, this paper utilizes WordPiece embeddings [16] as a form of dynamic word embedding in the stages of pre-training and fine-tuning PLMs.

Inspired by word embedding methods in transformers [26], the initial token for each sequence is consistently a special classification token ([CLS]). The final hidden state corresponding to this token is utilized as the aggregate sequence representation for classification tasks. Sentences are separated with a unique token ([SEP]). In addition to WordPiece embedding, segment embedding indicates which sentence the token belongs to, and position embedding indicates the relative position of the token in the sentence is applied to every token. In Fig. 3b, the token embedding is denoted as E_{token}, the segment embedding as E_{sn}, and the positional embedding for the i_{th} token in a sentence as E_i.

The word embeddings for prevailing PLMs, as depicted in Fig. 1a, rely on general English dictionaries. Such a method may not encapsulate the exact meanings of terminologies in the field of civil engineering, thereby potentially complicating the process of information extraction. The majority of text analysis research in construction management employs a general English dictionary for word embedding which often leading to less precise results. In contrast, our model will undergo self-supervised training on domain-specific corpora. As a result, texts within the CMS domain can be better represented by domain word embeddings, and consequently, PLM performance is expected to be enhanced. This procedure is demonstrated in the middle section of Fig. 2, dedicated to the re-training of PLMs on the domain corpus.

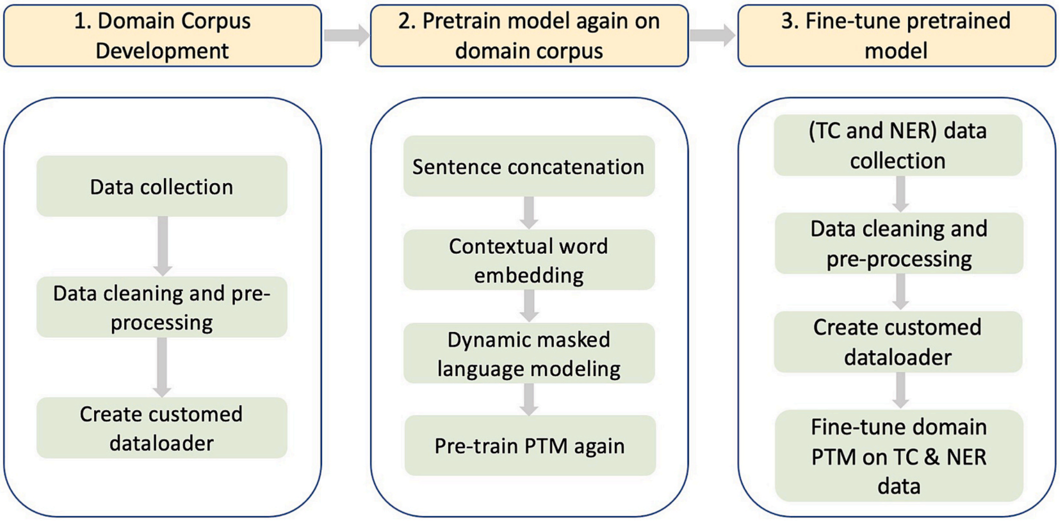
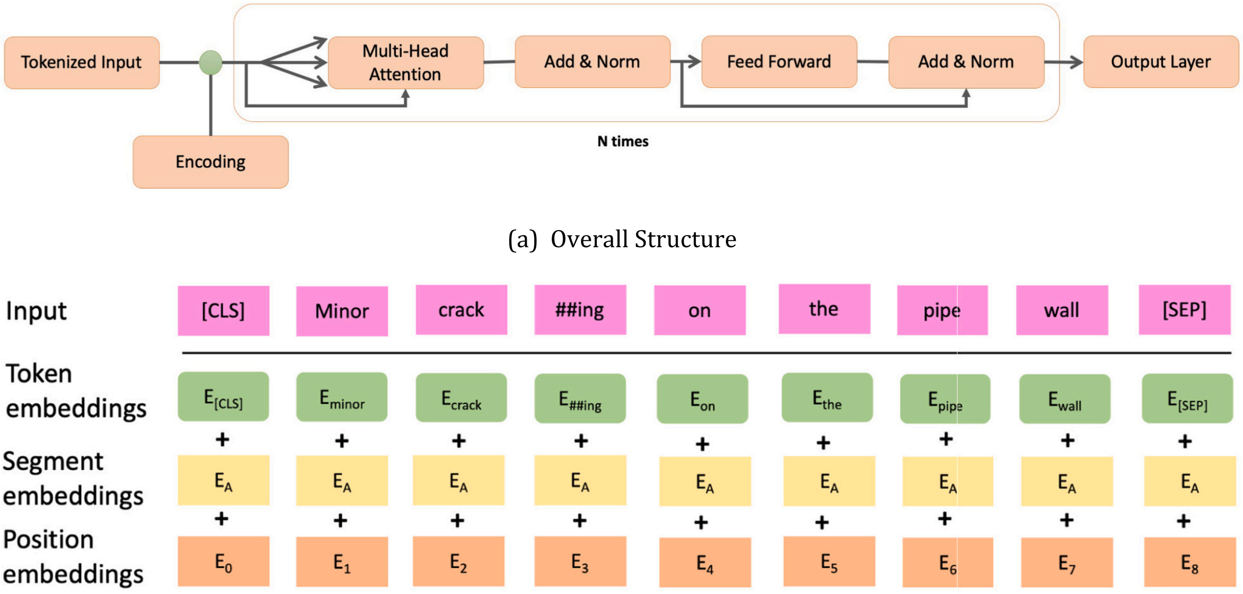

1. Domain Corpus Development
2. Pretrain model again on domain corpus
3. Fine-tune pretrained model
Data collection
Data cleaning and pre-processing
Create customed dataloader
Sentence concatenation
Contextual word embedding
Dynamic masked language modeling
Pre-train PTM again
(TC and NER) data collection
Data cleaning and pre-processing
Create customed dataloader
Fine-tune domain PTM on TC & NER data

Fig. 2. Detailed workflow of domain corpus-enhanced transfer learning methods.



(a) Overall Structure

(b) Encoding Procedure

Fig. 3. Transformer Encoder Model Architecture.

3.1.2. *Sentence concatenation*

In their study, Liu et al. [30] posited that the utilization of individual sentences during pre-training could negatively impact performance on downstream tasks. They hypothesized this detriment to be the result of the model's incapacity to learn long-range dependencies from isolated sentences. Furthermore, it is observed that the sentences contained within the CMS domain dataset exhibit an average length of approximately 30. Given this context, we presume that concatenating sentences to the maximum token length accepted by a pre-trained model (512 for BERT) might significantly enhance the training speed. Thus, we plan to conduct a comprehensive evaluation wherein the pre-training of the PLM is carried out on the CMS dataset both with and without sentence concatenation. Subsequently, we aim to compare the resultant outcomes to discern the impact of sentence concatenation on the model's performance and training efficiency.

Following data cleaning, sentence concatenation is executed in the following procedural steps: (1) Tokenize all sentences to yield a list of individual tokens. (2) Concatenate tokens into a single string, adhering to their original sequential order. (3) When the aggregate token count within the string surpasses the predefined maximum token length acceptable for the PLM, the string comprising the tokens that precede the sentence that exceeded the limit is appended as an independent data point. This systematic approach allows us to efficiently concatenate whole sentences while maintaining compliance with the constraints of the PLM's maximum token capacity.

3.1.3. *Dynamic masked language modeling*

In order to pre-train a PLM on domain-specific corpora, we initiate by randomly masking a certain percentage of the input tokens. This process, known as Masked Language Modeling (MLM), involves obscuring parts of the input text and training the model to predict these masked tokens. Subsequently, these masked tokens are predicted using the final hidden vectors, which are fed into an output softmax over the vocabulary, akin to a standard language model. As proposed in BERT [13], this method entails masking 15% of all WordPiece tokens with [MASK] token in each sequence randomly, then exclusively predicting the masked words rather than reconstructing the entire input. Though this strategy enables the attainment of a bidirectional pre-trained model, it may also lead to a mismatch between pre-training and fine-tuning due to the absence of the [MASK] token during the fine-tuning phase.

To alleviate this issue, BERT [13] introduced randomization into the [MASK] token. Using this method, our training data generator randomly selects 15% of the token positions for prediction. If the i^{th} token is chosen for prediction, the i^{th} token is replaced with (1) the [MASK] token 80% of the time, (2) a random token 10% of the time, and (3) the unchanged i^{th} token 10% of the time. Following this, cross-entropy loss will be employed to predict the original token, denoted by:

$$\mathscr{L}_{CE}(y,t) = -\sum_{k=1}^{K} \mathrm{t}_k log\ \mathrm{y}_k = -\boldsymbol{t}^T(log\ \mathbf{y})$$

where the log is applied element-wise, **t** signifies the true one-hot encoded label, K stands for the dimension of the label vector, and **y** refers to the predicted label.

The original BERT implementation carried out masking once during data preprocessing, resulting in a single static mask and a higher probability of overfitting. In contrast, Liu et al. [30] discovered that dynamic masking, where the masking pattern is generated every time a sequence is fed into the model, can enhance model performance on multiple datasets. Hence, we employ dynamic masking in the succeeding experiments.

The following steps outline the technical detailed procedure for pre-training the DL models mentioned above using the dynamic MLM approach:

1. Tokenization: The text is tokenized into three distinct tensors, namely input ids, token type ids, and attention mask. Token type ids is not required for MLM.
2. Creation of labels tensor: A labels tensor is created by replicating input ids. This tensor aids in calculating the loss and optimizing the model.
3. Masking of tokens in input ids: A random selection of tokens in input ids is masked for each batch of data.
4. Calculation of loss: The masked input ids and predicted labels tensors are processed through the BERT model to calculate the loss between them. The loss is computed as the discrepancy between the output

probability distributions for each output token and the true one-hot encoded labels.

It is essential to underscore that the MLM is a self-supervised task, thereby obviating the necessity for supplementary manual labeling endeavors. Following the pre-training of the BERT model on domain-specific corpora, the model parameters undergo optimization, thereby aligning them for potentially better results of NLP tasks within the context of CMS. Then, for diverse tasks, users can simply modify the last output layer and subsequently fine-tune the model, as depicted in Fig. 1c.

3.1.4. Pre-training with domain corpora procedure

The predominance of transformer-based models, which are frequently pre-trained on general-domain corpora [13], results in a data distribution that differs notably from that of the target domain. Consequently, it becomes compelling to consider an effective strategy that involves further pre-training these transformer-based models on data specific to the target domain. As illustrated in the middle segment of Fig. 2, the transfer learning method, when applied to pre-train transformer-based models using the domain corpus, is comprised of four main stages.

In the first stage, sentence concatenation is performed to produce more complex sentence structures as shown in Section 3.1.2. This step allows the model to develop a better understanding of how different sentences in the domain are related and contextually interconnected while reducing training time.

The second stage involves the vectorization of concatenated sentences in the CMS corpora built in Section 4.1, achieved by utilizing contextual word embeddings as shown in Section 3.1.1. The objective is to accurately encapsulate the specific semantic and syntactic attributes of the target domain.

The third stage entails the application of dynamic masking on the vectorized corpora as shown in Section 3.1.3, a method vital to enabling the model to better comprehend the context of each word and learn useful representations from both the preceding and succeeding tokens.

In the fourth and final stage, PLMs are pre-trained using CMS domain data following the successful execution of all preceding steps. The purpose here is to inculcate a comprehensive understanding of the CMS domain knowledge, thereby enhancing the performance of the pre-existing transformer-based models. The steps of this stage are shown in Fig. 4. Specifically for step 1, pre-training configurations encompass various facets such as the number of training epochs, learning rate, batch size, and other related parameters as delineated in Section 5.1.

Upon the completion of the pre-training phase for all PLMs using various pre-processed CMS domain corpora, the models that exhibit the least pre-training loss on the test dataset are selected separately for both BERT and RoBERTa. These models are then processed through the final phase, the fine-tuning process, which further enhances their performance within the context of the CMS domain.

3.2. Fine-tune pretrained model

The process of fine-tuning and prediction utilizing the domain-specific PLM for TC and NER tasks is illustrated in the right segment of Figs. 1 and 2. The BERT and RoBERTa models used in this study will serve as representative models for elucidation in this section. The entire process encompasses six principal steps:

1) Data collection for TC and NER, which involves erosion control structures inspection records dataset and information extraction dataset for ACC; 2) TC and NER data cleaning and pre-processing; 3) The creation of a custom dataloader to import data into PLMs. This process includes word embedding and encoding, where all tokens of the input sentence are initially turned into word embeddings, followed by the utilization of a PLM to encode the position and segment embeddings into contextual representations; 4) Fine-tuning of the domain-specific PLM on TC and NER dataset by obtaining prediction via output layer, where the contextual representations in the last hidden layer are then input into to obtain the prediction result; 5) Construct a set of baseline models that include a Bidirectional Gated Recurrent Unit (BiGRU) [34], a form of RNN [28], as well as a Logistic Regression (LR) with Gaussian Kernel [35]. When contrasted with other deep learning models, notably RNN-based models that have been widely employed for NLP tasks, transformer-based models have demonstrated substantial advancements in both language modeling performance and computational efficiency during model training [4]. To prove that transformer-based models exhibit particular prowess in handling long-term dependencies in textual content, which significantly bolsters their overall performance, these baseline models have been selected and compared to transformer-based models; 6) Execute a comparative analysis of the results produced by the original PLM, the domain-specific PLM, and the aforementioned baseline models. This comparison will provide a holistic overview of the efficacy and applicability of the various models in our specific context.

In step 4, a crucial procedure that requires emphasis is pre-processing. For baseline models, data pre-processing is executed instead of tokenization with the aid of the Natural Language Toolkit (NLTK) library and regular expressions. This pre-processing includes a series of operations such as conversion to lowercase, punctuation removal, handling of digit-word combinations, stopword removal, text rephrasing, stemming and lemmatization, and white space removal. On the other hand, transformer-based models necessitate less extensive and more automated pre-processing procedures, resulting in a more streamlined operation when compared to traditional machine learning and RNN-based models. The pre-processing phase for transformer-based models typically encompasses the addition of special tokens to differentiate sentences, padding sequences to a constant length, and the creation of an attention mask (which entails the generation of arrays populated with 0 s (representing pad tokens) and 1 s (indicating real tokens)).

The versatility of the transformer architecture permits PLMs to be effectively fine-tuned for a variety of downstream tasks by simply modifying the final output layer and the associated loss function. During this fine-tuning process, task-specific inputs and outputs are integrated into the PLM, facilitating an end-to-end adjustment of all model parameters. Fig. 5a illustrates the fine-tuning process for the TC task, where the input is a single sentence, and the output is the class label of that sentence. In this context, the figure underscores how the PLM discerns the overall thematic category or sentiment of the sentence, categorizing it into predefined classes. Conversely, Fig. 5b demonstrates the finetuning for the NER task. Here, the input remains a single sentence, but the output consists of labels for each semantic element within that sentence. Both figures collectively demonstrate the flexibility and effectiveness of PLMs in adapting to diverse NLP tasks, showcasing how the same underlying model architecture can be tailored to meet the specific requirements of different applications within construction management systems.

3.2.1. Performance evaluation metrics for fine-tuning tasks

The averaged F1 score, recognized by numerous researchers in the domain [11] [7], is employed as a yardstick for evaluating the performance of the deep learning models. Conceptually, the averaged F1 score is characterized as the harmonic mean of precision and recall, wherein

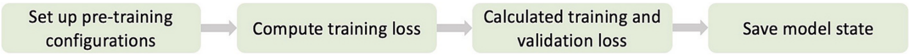
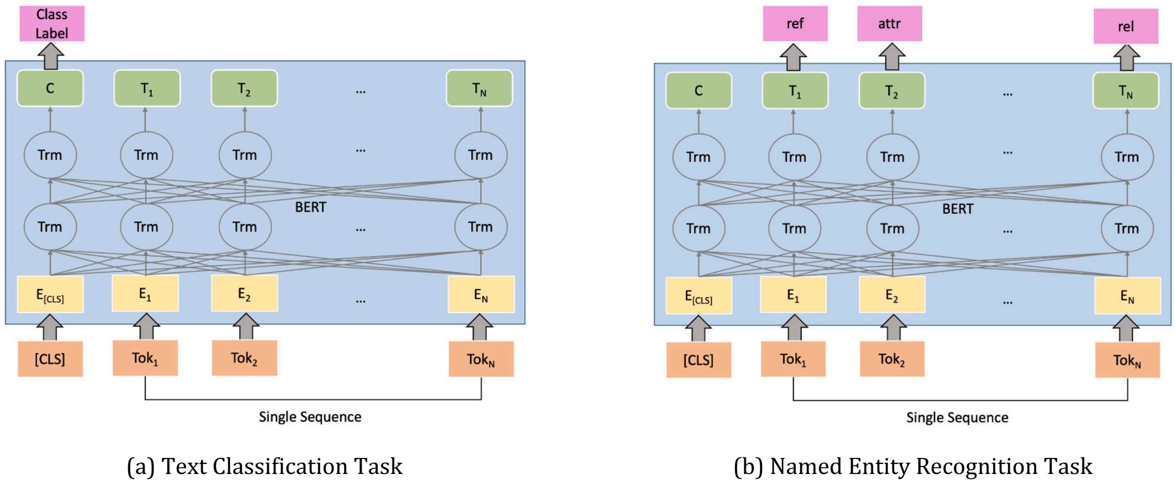

Set up pre-training configurations
Compute training loss
Calculated training and validation loss
Save model state

Fig. 4. Pre-training PTM workflow.



(a) Text Classification Task (b) Named Entity Recognition Task

Fig. 5. Fine-tuning process illustration.

an ideal F1 score equates to 1 while the least preferable score converges to 0. For multi-class and multi-label situations, the averaged F1 score constitutes the weighted average of the F1 scores of each class, with the weights being determined by the average parameter. The average F1 score is a reflection of the model's overall performance across all instances, which aligns with our goal to maximize the general accuracy in practical applications where the distribution of entities mirrors their real-world frequencies. Furthermore, the predominant classes in our dataset are of particular interest due to their higher practical relevance in the intended application of our NER system. Due to the complexity and diversity of the data, it's common to encounter imbalanced datasets. In such situations, relying solely on accuracy can be misleading since a high accuracy does not always translate to a good model, especially when there's a significant class imbalance. In these scenarios, metrics such as precision and recall become especially relevant. Therefore, the averaged F1 score is used because it effectively balances the trade-off between the two.

4. Data collection and data cleaning

In this research, we plan to engage an unlabeled dataset for the purpose of implementing self-supervised pre-training, as illustrated on the left side of Fig. 2. In an effort to provide a comprehensive analysis, four separate CMS datasets will be compiled as shown in Fig. 6. These datasets differ based on whether references have been omitted and whether sentence concatenation has been executed. Simultaneously, two labeled datasets will be employed, with the primary purpose being supervised finetuning, which is represented on the right side of Fig. 2. Subsequently, we will assess the effectiveness of domain corpora using both the Text TC and NER datasets.

4.1. *Domain corpora construction*

4.1.1. *Data collection procedure and rationality*

The first CMS domain corpus is constituted of academic publications pertinent to CMS, inclusive of scholarly journal papers, conference papers, articles, whitepapers, and a few books, reflecting a comprehensive range of academic discourse in the field. The rationale for this selection is grounded in the recognition that academic publications offer a rich source of specialized terminology and advanced concepts within the CMS domain, as emphasized in previous research [36]. These publications were primarily sourced from Google Scholar, selected for its extensive coverage and constant expansion of relevant academic literature [37]. Compared to other databases such as Scopus or Web of

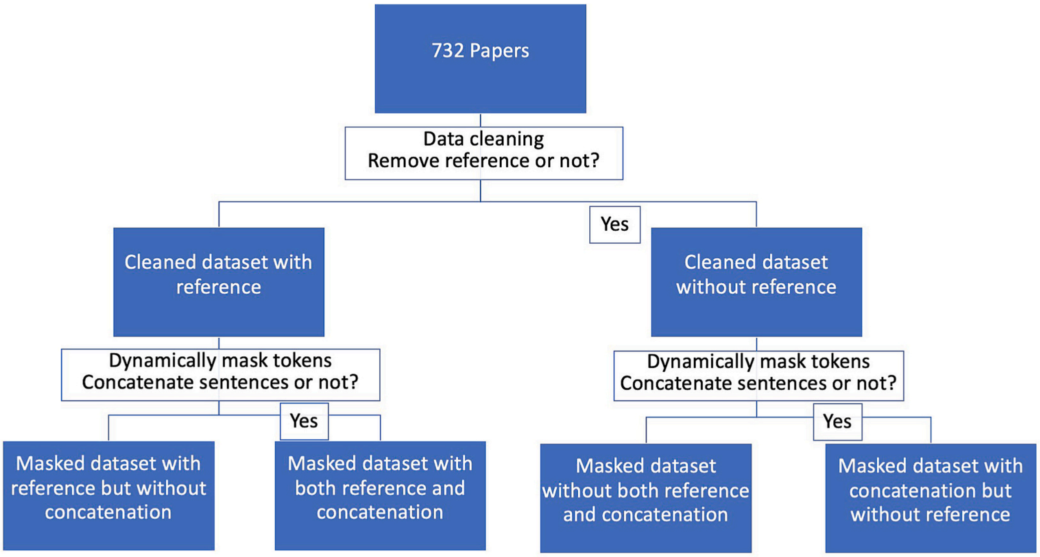

732 Papers
Data cleaning
Remove reference or not?
Yes
Cleaned dataset with reference
Cleaned dataset without reference
Dynamically mask tokens
Concatenate sentences or not?
Yes
Dynamically mask tokens
Concatenate sentences or not?
Yes
Masked dataset with reference but without concatenation
Masked dataset with both reference and concatenation
Masked dataset without both reference and concatenation
Masked dataset with concatenation but without reference

Fig. 6. Illustration of CMS domain datasets.

Science, Google Scholar provides an easier way of automatic information retrieval from experiments by the author and is therefore selected.

The academic publications were procured from the outcomes of the first 99 pages of Google Scholar using the keyword”construction management”. Out of 732 papers earmarked for training, 60 were discarded due to the inability to recognize the text in some of the older PDFs. In total, the corpus comprises 5.7 million words, and when references are excluded, the word count stands at 4.5 million. The total count of tokens is 7.7 million, and discounting the references (processed with BERT tokenizer), it is 5.8 million. It should be noted that more than 90% sentences are less than 40 tokens long. The corpus size, which comprises 5.7 million words, holds significance when compared to the in-domain dataset of 10 million Chinese characters in the earlier work by Zheng et al. [7], taking into account that individual English words are typically represented by multiple Chinese characters. The developed corpus offers a substantial volume of domain-specific data, essential for training effective PLMs in niche fields like CMS. The volume of this corpus aligns with established research underscoring the importance of corpus size for the successful training of language models, particularly in specialized domains where nuanced understanding is critical [38].

4.1.2. Data cleaning and pre-processing

CMS Domain corpus data is cleaned and pre-processed in the following steps:

1. Convert PDF to plain text (txt) automatically using Adobe Automation. This is essential to facilitate the subsequent data manipulation and analysis steps.
2. Remove website links because they generally do not contribute substantive content necessary for the pre-training process [39].
3. Only retain English text on the paragraph level using regular expression [40]. This serves to discard unrecognizable characters and any non-English text that could potentially disrupt the pre-training process.
4. Divide paragraphs into distinct sentences. Additionally, paragraphs lacking terminal punctuation marks are purged to exclude potential remnants of formulas and tables that are inherent to the PDF format.
5. Filter sentences that are too short to further remove any residual non-textual elements of formulas and tables, ensuring the quality and relevance of the data.
6. Filter references within the journal papers. These references may not contain useful context or topic-specific content beneficial for the pre-training process. To assess the impact of this step, datasets both with and without references were prepared for pre-training.
7. Removal of duplicate sentences. This measure ensures the uniqueness of each sentence, enhancing the diversity and coverage of the dataset for the pre-training task, aligning with the approaches in [13].

Following the comprehensive data cleaning procedure, exclusive of the reference filtration step, approximately 5% of the words in the original dataset have been excised. Subsequently, during the process of filtering references, a further reduction of approximately 20% in word count is observed. This signifies that the reference sections constituted a substantial portion of the original word count, reinforcing the necessity for their careful scrutiny and selective filtration to maintain the relevance and quality of the corpus for pre-training tasks. The domain corpora released as part of this study contain both uncleaned and unprocessed data and cleaned data. The rationale for this decision stems from the understanding that a universally optimal procedure for various pre-training and fine-tuning tasks does not currently exist. Consequently, providing the raw data allows researchers the flexibility to adopt or design appropriate preprocessing methods tailored to their specific task requirements. A repository containing the dataset was established on GitHub at https://github.com/zhongyunshun/domain-corpora.

4.2. CMS domain dataset construction

4.2.1. Text classification dataset construction

Assessing conditions of erosion control structures is fundamental to monitoring and maintenance of existing erosion control structures along Toronto and Region Conservation Authority (TRCA)’s rivers and valleys that protect public greenspace, park amenities, and municipal infrastructure in the Great Toronto Area (GTA) [41]. Therefore, the analysis of inspection records can offer invaluable insights into maintenance needs, structural stability, and investment requirements. In this research, the TC dataset comprises records of inspections from 1950 to 2021 derived from TRCA’s Erosion Risk Management Program. The objective is to apply Transformer models pre-trained on CMS domain corpora to predict the structural condition of the erosion control structures, an application particularly suited due to the complexity of the unstructured data and the need for nuanced interpretation.

The dataset encompasses both unstructured data—such as inspection records detailing structural conditions and maintenance priority rationales—and structured data, including elements like overall condition, structure stability, maintenance priority, location, inspection time, and required maintenance investment. The inspection records pertaining to structural conditions serve as inputs, while the corresponding structural overall conditions function as labels in the dataset. The dataset encompasses a total of 30,637 inspection records. A detailed data cleaning process was implemented to enhance prediction accuracy, encompassing the removal of duplicate or irrelevant observations, the rectification of structural errors such as typographical errors and inappropriate capitalization, filtration of undesired outliers, handling of missing data, and conversion to the lower casing. This systematic approach to data cleaning and classification contributes to the reliability and accuracy of the ensuing analysis and predictions.

Following the data cleaning process, the resultant dataset includes 7199 structures categorized as being in excellent condition, 12,022 structures in good condition, 6700 structures in fair condition, and 2796 structures classified as being in poor condition. Fig. 7 shows the distribution of structural conditions by region. Toronto has the most erosion control structures, followed by Peel, York, and Durham regions. Approximately 50% of the erosion control structures are categorized as being in good condition. The quantities of structures in both excellent and fair conditions are nearly equivalent. A mere count of approximately 3000 structures are classified as being in poor condition. An example of an erosion control structure in excellent condition is “The structure and slope behind the structure appear to be stable. Minor Displacement of structure in the D/S portion of the structure. No other deficiencies were observed at the time of inspection.” An example of an erosion control structure in poor condition is”Cribwall armors the left bank along a straight section. Field staff unable to inspect the entire length of structure due to major sediment buildup on top of structure. Moderate debris buildup throughout entire length.”

In summary, this dataset construction paves the way for an innovative application of Transformer models pre-trained on CMS domain corpora to analyze and predict the condition of erosion control structures, a task of significant importance for the safety and sustainability of the GTA region. By converting complex inspection records into a format suitable for deep learning, this research offers a practical solution for enhancing the monitoring and maintenance of infrastructure, with potential applications extending to other civil engineering contexts.

4.2.2. Named entity recognition dataset construction

In this study, we leverage an information extraction dataset comprising building code sentences designed for ACC. This NER task is vital for ensuring that various construction elements are in alignment with legal and regulatory standards. The dataset, specifically curated for this task, serves to identify, classify, and extract specific entities, rules, and attributes related to construction compliance, thus automating the complex and time-consuming process of manual compliance checking.

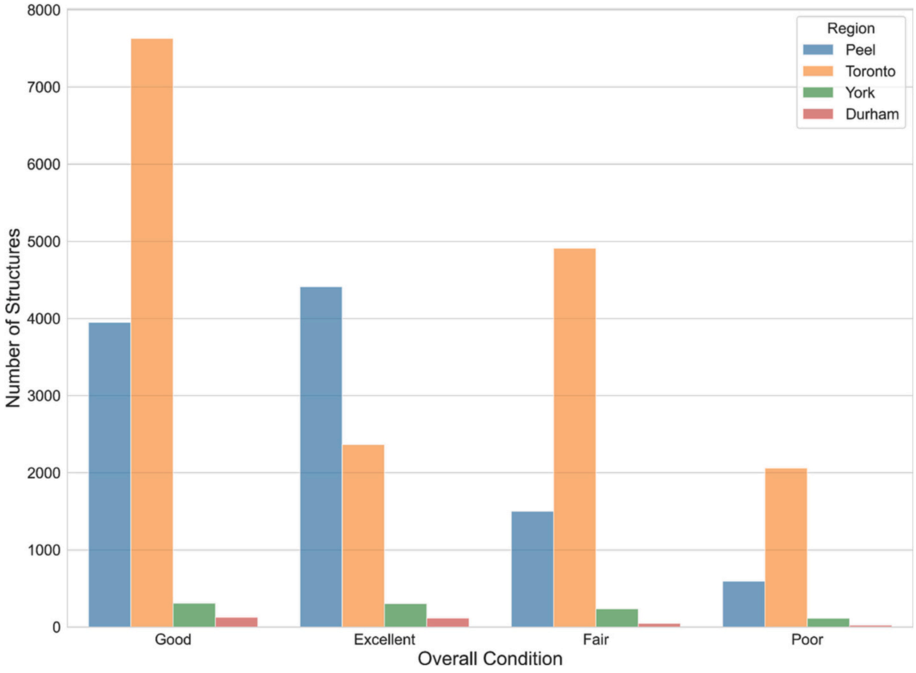


Fig. 7. Number of structures in each condition by region.

These sentences have been annotated with relevant semantic and syntactic information elements, as proposed by Zhang and El-Gohary [11], for the purpose of the NER task.

The dataset incorporates eight critical semantic information elements, a.k.a. part-of-speech tags, detailed in Table 1, which include the subject, compliance checking attribute, deontic operator indicator, comparative relation, quantity value, quantity unit, subject relation, and reference. These elements are fundamental in understanding and interpreting building codes, as they encapsulate the nuanced requirements and constraints that must be adhered to in construction projects. The syntactic information elements, though essential for the formation of grammatically correct building code sentences, do not contribute directly to the articulation of the building code requirement's semantics. These syntactic elements encompass conjunctions (e.g., "and"), disjunctions (e.g., "or"), negations (e.g., "not"), and terminal punctuation marks (e.g., ".", ",", ":", and ";"). A comprehensive list of all information elements is provided in Table 1, and a graphical representation of the dataset distribution is depicted in Fig. 8 [11]. In the figure, the X-axis shows each part-of-speech tag and the Y-axis shows the number of values for each tag.

Table 1
Semantic information elements for representing requirements for compliance checking [11].

Semantic information element	Definition
Subject	This represents an ontology concept of an entity (e.g., building element) subject to a specific requirement.
Compliance checking attribute	This ontology concept represents a distinct characteristic of a"subject" that is checked for compliance
Deontic operator indicator	A term or phrase that deontic type of the requirement (i.e., obligation, permission, or prohibition)
Comparative relation	A term or phrase for quantitive comparisons such as"greater than or equal to","greater than","less than or equal to","less than", and"equal to"
Quantity value	A numerical value that defines the quantity
Quantity unit	The unit of measure associated with a"quantity value"
Subject relation	A term or phrase that clarify the type of relation between two subjects, a subject and an attribute, or a subject or an attribute and a quantity
Reference	A term or phrase indicating references to a chapter, section, document, table, or equation within a building-code sentence
Conjunction	A term that unifies two or more sentences, phrases, or clauses together, thereby forming a conjunctive statement.
Disjunction	A term indicating alternatives among two or more sentences, phrases, or clauses.
Negation	A logical operation that inverts the truth value of a statement, phrase, or clause.
Terminal punctuation marks	A symbol used to signify the end of a complete thought or statement. They mark the conclusion of a sentence, thus delineating its boundary.

The utilization of Transformer models in this NER task has considerable advantages. With their self-attention mechanism, Transformers can accurately capture long-range dependencies and intricate relationships within building code sentences. This leads to a more effective identification and extraction of the requisite entities and attributes. Moreover, by pre-training on a domain-specific corpus, Transformer models can gain an enhanced understanding of the specific terminology and context of construction management, ensuring both accuracy and efficiency in ACC.

In addition, Fig. 9 shows example sentences from the International Building Code (IBC), International Energy Conservation Code (IECC), and Americans with Disabilities Act (ADA) Standards, and how the sentences are annotated using the proposed semantic and syntactic information elements. These examples further illustrate the complexity and diversity of building code sentences, emphasizing the need for advanced machine learning techniques such as Transformer models to tackle the ACC task.

5. Experiments, results, and discussion

To explore the effectiveness of domain corpora and transfer learning methodologies for domain-related NLP tasks, specifically TC and NER within the CMS domain, we conduct several experiments in this section.

During the pre-training phase, we partition the domain corpus dataset into a 97:3 training-to-test ratio. This ratio is chosen based on the large size of the dataset, where a 3% test set is deemed adequate for a robust evaluation.

In the subsequent fine-tuning phase, the TC and NER datasets are divided into training, validation, and test sets in an 80:10:10 ratio. This

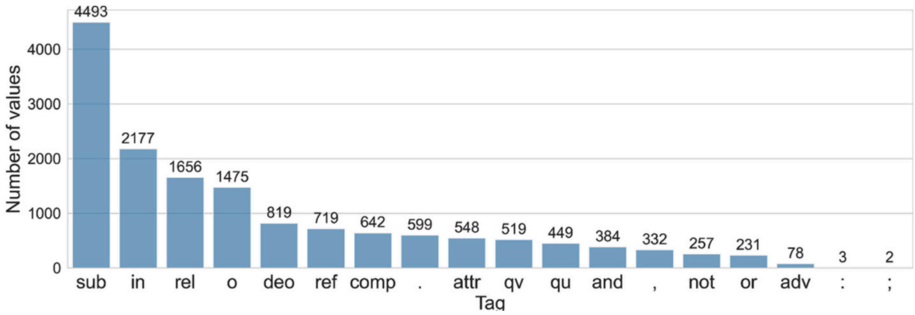


Fig. 8. Number of values for each tag.

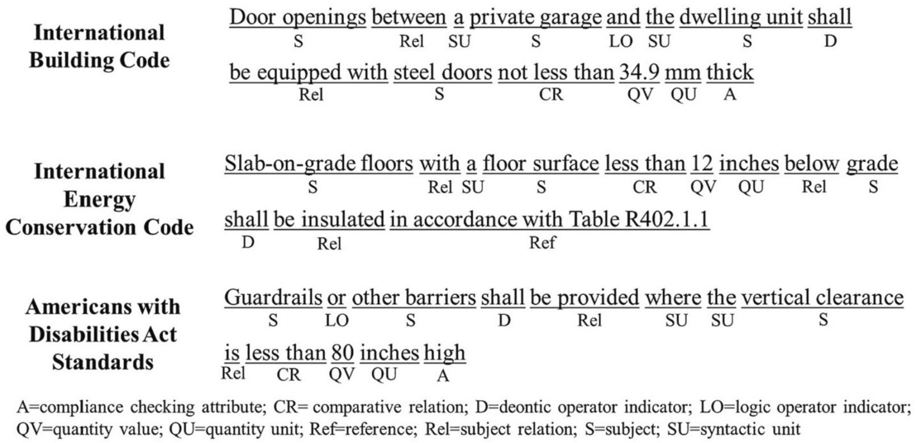

International Building Code
Door openings between a private garage and the dwelling unit shall
S Rel SU S LO SU S D
be equipped with steel doors not less than 34.9 mm thick
Rel S CR QV QU A
International Energy Conservation Code
Slab-on-grade floors with a floor surface less than 12 inches below grade
S Rel SU S CR QV QU Rel S
shall be insulated in accordance with Table R402.1.1
D Rel Ref
Americans with Disabilities Act Standards
Guardrails or other barriers shall be provided where the vertical clearance
S LO S D Rel SU SU S
is less than 80 inches high
Rel CR QV QU A
A=compliance checking attribute; CR= comparative relation; D=deontic operator indicator; LO=logic operator indicator; QV=quantity value; QU=quantity unit; Ref=reference; Rel=subject relation; S=subject; SU=syntactic unit

Fig. 9. Example building-code sentences annotated with the proposed syntactic and semantic information elements [11].

division is executed using a random data division approach to ensure that the datasets represent a wide range of examples and to avoid any potential bias. This setting is the same as prior work by Zheng et al. [7]. This division is executed using a random data division approach to ensure that the datasets represent a wide range of examples and to avoid any potential bias. Here, the training set is utilized to train and iteratively update PLMs. Concurrently, the validation set serves to evaluate model performance and assist in identifying the optimal combination of hyperparameters, as well as the best model variant. Lastly, the test set is employed for the final evaluation of model performance.

5.1. Pre-training configuration

This experiment aims to investigate the performance of different PLMs on domain-specific datasets. In the phase of PLMs pre-training on the domain corpora, we take into consideration four distinct types of domain corpora. These include 1) domain corpora devoid of reference and sentence concatenation (D); 2) domain corpora without reference, yet incorporating sentence concatenation (DS); 3) domain corpora incorporating reference but without sentence concatenation (DR); 4) domain corpora employing both reference and sentence concatenation (DSR). Two PLMs, BERT [13] and RoBERTa [30], have been meticulously chosen for this task.

5.1.1. BERT model

The BERT base model [13], a seminal figure in the landscape of PLMs, utilizes the transformer encoder. Upon its advent, it surpassed its competitors over a broad spectrum of tasks. BERT's commendable capacity to discern context and produce semantically meaningful embeddings has earned it significant appreciation. The model architecture is a multi-layer bidirectional Transformer encoder based on Vaswani et al. [13,26]. It has stacked self-attention and point-wise, fully connected layers. The model is composed of a stack of $N = 12$ identical layers (Transformer blocks). Each layer has two sublayers: one is a multi-head self-attention mechanism, and the other is a simple, position-dependent fully connected feed-forward network. Following a residual connection [42] around each sub-layer, we normalize the layers by applying the function LayerNorm(x + Sublayer(x)) [43], where Sublayer(x) is the function implemented by the sub-layer itself. We denote the hidden size as H = 768, the number of self-attention heads as A = 12, and the embedding length E = 512. The total parameters are 110 M.

5.1.2. RoBERTa model

The RoBERTa [30] model is an enhanced iteration of the BERT model with the same architecture as BERT, characterized by its extended training duration, larger batch sizes, and more comprehensive data coverage. It eschews the next sentence prediction objective in favor of training on lengthier sequences, while also dynamically altering the masking pattern applied to the training data. It has delivered superior performance across a wide range of datasets compared to BERT model and has demonstrated its adeptness and flexibility in managing a diverse range of NLP tasks [30].

5.1.3. Hyperparameters and training

Four BERT and RoBERTa models pre-trained on CMS domain corpora are obtained, as listed in Table 2.

Table 2
PLMs with different CMS domain corpora.

Base model/ Corpus	D	DS	DR	DSR
BERT RoBERTa	BERT-D RoBERTa-D	BERT-DS RoBERTa-DS	BERT-DR RoBERTa-DR	BERT-DSR RoBERTa-DSR

The aforementioned models are trained with a maximum of 5 epochs, which exceeds the epoch count used in the original training of the BERT model (namely 4 epochs). The exploration of diverse learning rates and batch sizes is performed via grid search, encompassing initial learning rates of 1×10^{-3}, 5×10^{-4}, 3×10^{-4}, 1×10^{-4}, 5×10^{-5}, 3×10^{-5}, and 1×10^{-5}; weight decay of 0.001, 0.003, 0.005, 0.008, 0.01, 0.03, 0.05, 0.08, and 0.1; and warmup steps of 20, 50, 80, 100, 150, 200, fostering an optimal balance between model adaptability and overfitting control. Furthermore, a variety of maximum token lengths (128, 256, and 512) and batch sizes (16, 32, and 64) are considered. Upon the completion of this training process, an initial learning rate of 1×10^{-4} is selected, along with maximum token lengths of 512, a batch size of 16, a weight decay of 0.01, and a warmup steps of 200. These parameter choices emerge from our experimentation as the most conducive to achieving robust performance in our specific modeling context. For optimization, the AdamW algorithm [44] is employed with all other parameters kept at their default settings. The cross-Entropy loss function is employed to measure the difference between the predicted probability distribution for the masked tokens and the ground truth, aligning with the original BERT and RoBERTa model.

5.2. Pre-training results

In Table 3 and Table 4, the training and evaluation loss associated with the domain corpus as well as training time for all BERT and RoBERTa models are displayed. It can be seen that the utilization of sentence concatenation strategies has yielded significant efficiency benefits, reducing the requisite training time to a sixth of the original duration while simultaneously lowering evaluation loss by 14% and 27% for BERT and RoBERTa models. The excision of reference citations from the training corpus has led to an increase in evaluation loss by 4%, but it reduces training time by 28% for BERT model. On the contrary, the excision of reference citations from the training corpus slightly reduces evaluation loss and also reduces training time by 29%. The potential explanation for this observation could be the role that reference citations play in the dataset, acting more akin to noise. Consequently, their removal does not substantially impact the pre-training loss.

Based on the outcomes procured from the pre-training phase, the selection of models for subsequent fine-tuning tasks focused on two variants: the BERT model, pre-trained on domain-specific data with both reference citations and sentence concatenation (BERT-DSR), and the RoBERTa model, pretrained on domain-specific data but with references removed and sentence concatenation implemented (RoBERTa-DS). This decision was informed by the outstanding performance metrics demonstrated by these models, compounded by the aforementioned benefits of sentence concatenation.

5.3. Fine-tuning configuration

This experiment aims to investigate the performance of PLMs pre-trained on CMS domain corpora for TC and NER tasks. BERT-DSR and RoBERTa-DS are selected for TC and NER tasks. After being pre-trained on domain corpora, domain-specific PLMs are fine-tuned with a maximum of 10 epochs, which exceeds the epoch count typically used in downstream datasets (usually 2–4 epochs). The exploration of diverse learning rates and batch sizes is performed via grid search, encompassing initial learning rates of 1×10^{-3}, 5×10^{-4}, 3×10^{-4}, 1×10^{-4}, 5×10^{-5}, 3×10^{-5}, 1×10^{-5}, 5×10^{-6}, 3×10^{-6}, and 1×10^{-6}; weight decay of 0.001, 0.003, 0.005, 0.008, 0.01, 0.03, 0.05, 0.08, and 0.1; and warmup steps of 20, 50, 80, 100, 150, 200, fostering an optimal balance between model adaptability and overfitting control. Furthermore, a variety of maximum token lengths (128, 256, and 512) and batch sizes (16, 32, and 64) are considered. Upon the completion of this training process, an initial learning rate of 1×10^{-5} is selected, along with weight decay of 0.05, warmup steps of 80, maximum token lengths of 512, and a batch size of 32. These parameter choices emerge from our experimentation as the most conducive to achieving robust performance in our tasks. For optimization, the AdamW algorithm [44] is employed with adam beta2 of 0.001, 0.003, 0.005, 0.008, 0.01, 0.03, 0.05, 0.08, 0.1. Adam beta2 of 0.995 is selected after grid search. The control group in this experiment comprises models identical to the test group but excludes the pre-training on domain-specific corpora; that is to say, these control models are exclusively pre-trained on the general corpora (original BERT and RoBERTa).

Table 3
Domain-specific BERT model pre-training results.

Results/ Model	BERT-D	BERT-DS	BERT-DR	BERT-DSR
Training loss	2.363	**1.658**	2.359	1.879
Evaluation loss	2.089	1.910	2.009	**1.679**
Pre-training time (hrs)	2.54	**0.44**	3.54	0.59

Table 4
Domain-specific RoBERTa model pre-training results.

Results/ Model	RoBERTa-D	RoBERTa-DS	RoBERTa-DR	RoBERTa-DSR
Training loss	2.166	1.499	2.128	**1.496**
Evaluation loss	1.781	**1.297**	1.788	1.305
Pre-training time (hrs)	2.79	**0.49**	3.90	0.65

5.4. Text classification results

The evaluation of the various pre-trained models on the TC dataset is detailed in Table 5 and Fig. 10. All PLMs that are pre-trained on domain-specific corpora exhibit superior performance compared to those trained only on generic corpora. Among the investigated models, the domain-specific RoBERTa, employing sentence concatenation, recorded the highest F1-score of 0.754. On average, an improvement of 5.9% in the F1-score was observed. This increase is distributed between BERT and RoBERTa with enhancements of 5.6% and 6.1% respectively, rendering the F1-score of RoBERTa marginally superior to that of BERT. Compared to baseline models (LR and BiGRU), all transformer-based models have significantly higher F1-score. For instance, the F1-score of the domain-specific RoBERTa outperformed the kernelized LR by 16.6% and the BiGRU by 10.8%.

The observed performance improvements can be attributed to the domain-specific pre-training, which significantly enhances the models' comprehension of CMS-specific terminology and contextual nuances. By training on a corpus rich in construction management terminology, the models develop a more refined understanding of the semantic relationships and contextual usage specific to this domain. This specialized training enables the models to more accurately classify texts in CMS, as they can better discern subtle differences in meaning that generic models might overlook. These observed advancements underscore the potential utility of domain-specific PLMs when employing Deep Learning models for TC tasks within the CMS domain.

It merits mentioning that our TC task is predicting structural conditions of erosion control structures, a task that aligns more closely with

Table 5
Performance of different domain-specific and original models on text classification dataset.

Model	Precision	Recall	F1 score
Original BERT-DSR	0.694	0.689	0.689
Domain BERT-DSR	0.751	0.750	0.745
Original RoBERTa-DS	0.702	0.709	0.693
Domain RoBERTa-DS	**0.753**	**0.758**	**0.754**
Kernelized Logistic Regression	0.601	0.578	0.588
BiGRU	0.624	0.669	0.646

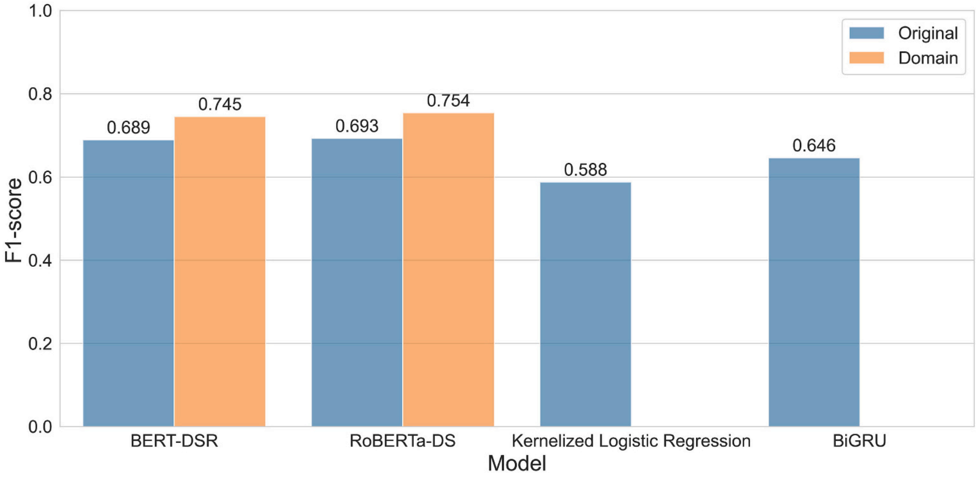


Fig. 10. Barplot of performance of different domain-specific and original models on text classification dataset.

infrastructure asset management in the AEC domain rather than construction management. However, our corpus is in the CMS domain. This observation suggests that our domain-specific pre-training approach can enhance performance across a wider spectrum in the AEC domain.

5.5. Named entity recognition results

Similar to the TC task, the evaluation of the various pre-trained models on the NER dataset is detailed in Table 6 and Fig. 11, reveals that domain-specific pre-training yields better performance. Among the investigated models, the domain-specific RoBERTa, employing sentence concatenation, recorded the highest F1-score of 0.956. On average, an improvement of 8.5% in the F1-score was observed. This increase is distributed between BERT and RoBERTa with enhancements of 8.8% and 8.1% respectively, rendering the F1-score of RoBERTa marginally superior to that of BERT. Recalls for all models except BiGRU are higher than precisions, leading to a high number of false positives. For the NER task, Precision ensures that the entities the model identifies are indeed correct and recall assesses the model's ability to find all instances of a specific entity within the text. For instance, in identifying the"compliance checking attribute" entity in the sentence"Dwelling unit shall be equipped with steel doors not less than 34.9 mm thick.", high recall would mean that the model effectively identifies most or all occurrences of"compliance checking attribute" such as"thick" in the dataset. However, this might be at the expense of incorrectly tagging other types of entities as"compliance checking attribute". Compared to baseline models (LR and BiGRU), all transformer-based models have significantly higher F1-score. For instance, the F1-score of the domain-specific RoBERTa outperformed the kernelized LR by 24.2% and the BiGRU by 20.3%.

Table 6
Performance of different domain-specific and original models on named entity recognition dataset.

Model	Precision	Recall	F1 score
Original BERT-DSR	0.826	0.892	0.858
Domain BERT-DSR	0.943	0.948	0.946
Original RoBERTa-DS	0.866	0.883	0.875
Domain RoBERTa-DS	**0.954**	**0.957**	**0.956**
Kernelized Logistic Regression	0.742	0.688	0.714
BiGRU	0.794	0.764	0.753

This improvement is particularly significant given the complexity of NER tasks in the CMS domain, where accurate identification of technical terms and specific jargon is crucial. The domain-specific pretraining equips the models with a nuanced understanding of CMS-related texts, enabling them to more effectively differentiate and label entities specific to construction management. These observed advancements underscore the potential utility of domain-specific PLMs when employing Deep Learning models for NER tasks within the CMS domain.

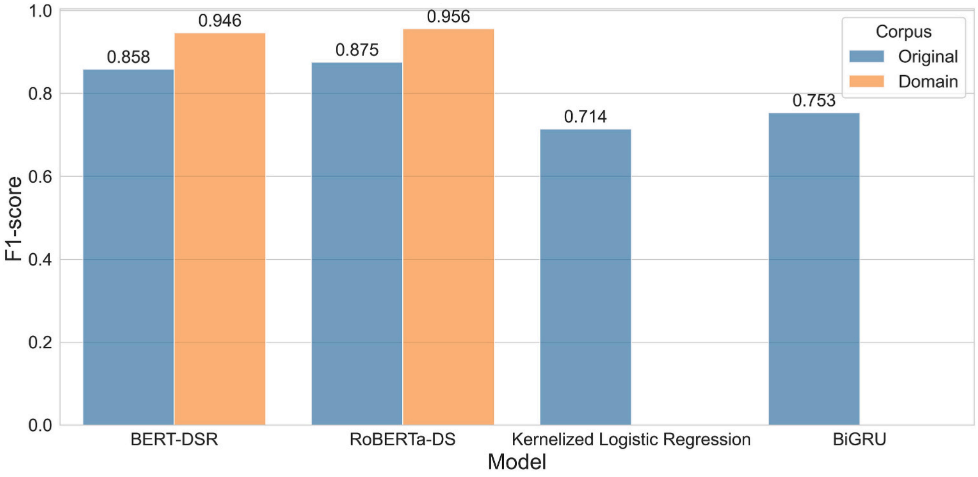


Fig. 11. Barplot of performance of different domain-specific and original models on named entity recognition dataset.

6. Discussion and conclusion

This work is primarily characterized by its contribution to the development of the first CMS domain corpus, and by its advancement of DL-based transfer learning methodologies for several NLP tasks within the CMS domain. We have comprehensively applied four datasets to two popularly used PLMs during pretraining to enhance the performance of DL models in CMS downstream tasks without manual annotation. Consequently, the best model for each PLM during pretraining is selected for fine-tuning on the TC and NER datasets. The domain corpus and pre-training framework we propose may also be applied effectively in other civil engineering contexts. Compared to previous efforts, this research specifically contributes in three ways.

First, this study presents the first publicly accessible domain corpus designed specifically for transfer learning within the CMS domain for pre-training domain-specific PLMs. Our approach leverages advanced Transformer-based architectures and transfer learning methodologies to harness both general and domain-specific information for NLP tasks within the CMS domain. We have developed four corpora employing varied data cleaning and pre-processing techniques to systematically analyze the impact of domain corpora and transfer learning techniques. Our work provides a means of leveraging domain semantics for word and sentence representation, thereby facilitating the analysis of highly technical, domain-specific texts within the construction management realm. Additionally, our approach offers a procedure for obtaining, cleaning, and pre-processing text data from PDF files, an asset that is of significant value and scarcity within the CMS domain.

Second, our research has established a highly automated, end-to-end pipeline for pre-training and fine-tuning domain PLMs with minimal pre-processing and hyperparameter tuning. This stands in contrast to existing machine learning and recurrent neural network-based methods, which typically require heavy pre-processing to achieve satisfactory results. Utilizing this pipeline, we have developed eight models during the pre-training stage and four domain-specific models for NER and TC tasks in fine-tuning stage. By changing only the last output layer of the models, they are capable of handling various types of datasets for NLP tasks within the CMS and related domains.

Third, this study is the inaugural endeavor to employ domain pre-trained transformer-based models in text and knowledge analytics to support ACC as well as infrastructure condition prediction. Our proposed models are able to handle diverse datasets within the CMS domain, and even some datasets within the broader AEC domain as shown in Section 5.4. The proposed models can not only consider the context of the word but also take different meanings of a word in different contexts into account, especially within the CMS domain.

This project aims to address the current lack of a comprehensive and dedicated corpus and language model for the IMS domain, which will enable more accurate and efficient natural language processing tasks such as text classification, information extraction, and named entity recognition. This research undertakes a systematic examination of how domain-specific corpora can potentially enhance the performance of deep learning models deployed for TC and NER in the CMS domain. Corpora in the CMS domain are developed and made publicly accessible for further exploration and utilization. In particular, sentence concatenation can reduce the training time to one-sixth of the original training time. The advantages of the developed domain corpora and domain-specific contextual word-embedding based DL models (such as BERT and RoBERTa) are demonstrated through TC and NER tasks.

For all evaluated NLP tasks, the PLMs pre-trained on domain-specific corpora consistently outperform those pre-trained on general corpora. For TC tasks, the application of domain corpora enhances the performance of PLMs pre-trained on general corpora, delivering a substantial increase of 5.9% in the F1 score. For NER tasks, the domain corpora similarly bolster the effectiveness of PLMs pretrained on general corpora, registering a marked improvement of 8.5% in the F1 score. Across all tested NLP tasks, PLMs pre-trained on both domain-specific and general corpora outperform static word-embedding based baseline models, with an average improvement F1-score of 10.3% for TC and 17.5% for NER tasks, thus highlighting the superior capabilities of the transformer architecture. In summary, this research project advances our understanding and application of PLMs within the CMS domain, offering new avenues for enhancing the performance of NLP tasks in this field.

While the advancements presented in this research are noteworthy, there remain considerable opportunities for future investigations that could potentially be of significant value to both academia and industry. Several potential directions for future work are outlined here. Firstly, the domain corpus constructed in this study could be further enriched by incorporating additional text resources such as Wikipedia entries and regulatory documents that are relevant to the CMS domain. However, the nature of Wikipedia content necessitates careful consideration and verification of the information to ensure its accuracy and relevance to the CMS domain. Such an expansion could potentially improve the representation of domain-specific knowledge in the corpus and thereby enhance the performance of domain-specific PLMs. Secondly, investigations into the impacts of dataset characteristics on model performance may also be performed. This could involve experiments designed to understand how factors such as the type, quality, and size of datasets influence the accuracy and robustness of the pre-trained models. Additionally, a more comprehensive exploration of model parameters and hyperparameters should be undertaken in future studies. This would provide a more granular understanding of the effects of different configuration choices on model performance and could identify optimal settings for various tasks within the CMS domain. Lastly, while this work has examined word embedding techniques and specific pretraining strategies, there are additional transfer learning methodologies that may prove beneficial within the CMS domain. Future research could explore approaches such as multitask training or the utilization of more recent PLMs. These strategies could potentially offer more efficient ways to leverage the rich patterns captured in large-scale general-domain corpora for tackling downstream tasks in the CMS domain.

In conclusion, while our study has paved the way toward an improved understanding and utilization of PLMs within the CMS domain, there remains a myriad of unexplored avenues that hold great potential for further enhancing the performance of NLP tasks in this field.

Declaration of generative AI

During the preparation of this work, the author(s) used GPT3.5 in order to correct grammar and spelling mistakes. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

CRediT authorship contribution statement

Yunshun Zhong: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Sebastian D. Goodfellow:** Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset for pre-training large language models (LLMs) are available in github. However, the datasets for fine-tuning pre-trained

LLMs are not available due to confidentiality agreement.

References

[1] Wu Chengke, Xiao Li, Yuanjun Guo, Jun Wang, Zengle Ren, Meng Wang, Zhile Yang, Natural language processing for smart construction: current status and future directions, Autom. Constr. 134 (2022) 104059, https://doi.org/10.1016/j.autcon.2021.104059.

[2] Zhe Zheng, Yu-Cheng Zhou, Lu Xin-Zheng, Jia-Rui Lin, Knowledge-informed semantic alignment and rule interpretation for automated compliance checking, Autom. Constr. 142 (2022) 104524, https://doi.org/10.1016/j.autcon.2022.104524.

[3] Weili Fang, Hanbin Luo, Xu Shuangjie, Peter E.D. Love, Lu Zhenchuan, Cheng Ye, Automated text classification of near-misses from safety reports: an improved deep learning approach, Adv. Eng. Inform. 44 (2020) 101060, https://doi.org/10.1016/j.aei.2020.101060.

[4] Ruichuan Zhang, Nora El-Gohary, Transformer-based approach for automated context-aware ifc-regulation semantic information alignment, Autom. Constr. 145 (2023) 104540, https://doi.org/10.1016/j.autcon.2022.104540.

[5] Arne Deloose, Glenn Gysels, Bernard De Baets, Jan Verwaeren, Combining natural language processing and multidimensional classifiers to predict and correct cmms metadata, Comput. Ind. 145 (2023) 103830, https://doi.org/10.1016/j.compind.2022.103830.

[6] Fahad Ul Hassan, Tuyen Le, Xuan Lv, Addressing legal and contractual matters in construction using natural language processing: a critical review, J. Constr. Eng. Manag. 147 (9) (2021) 03121004, https://doi.org/10.1061/(ASCE)CO.1943-7862.0002122.

[7] Zhe Zheng, Lu Xin-Zheng, Ke-Yin Chen, Yu-Cheng Zhou, Jia-Rui Lin, Pretrained domainspecific language model for natural language processing tasks in the aec domain, Comput. Ind. 142 (2022) 103733, https://doi.org/10.1016/j.compind.2022.103733.

[8] Peng Zhou, Nora El-Gohary, Domain-specific hierarchical text classification for supporting automated environmental compliance checking, J. Comput. Civ. Eng. 30 (4) (2016) 04015057, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000513.

[9] Min-Yuan Cheng, Denny Kusoemo, Richard Antoni Gosno, Text mining-based construction site accident classification using hybrid supervised machine learning, Autom. Constr. 118 (2020) 103265, https://doi.org/10.1016/j.autcon.2020.103265.

[10] Wu Lang-Tao, Jia-Rui Lin, Shuo Leng, Jiu-Lin Li, Hu. Zhen-Zhong, Rule-based information extraction for mechanical-electrical-plumbing-specific semantic web, Autom. Constr. 135 (2022) 104108, https://doi.org/10.1016/j.autcon.2021.104108.

[11] Ruichuan Zhang, Nora El-Gohary, A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking, Autom. Constr. 132 (2021) 103834, https://doi.org/10.1016/j.autcon.2021.103834.

[12] Xu Xin, Hubo Cai, Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure, Adv. Eng. Inform. 48 (2021) 101288, https://doi.org/10.1016/j.aei.2021.101288.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, 2018, https://doi.org/10.48550/arXiv.1810.04805.

[14] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., Improving language understanding by generative pre-training, 2018. https://www.mikecaptain.com/resources/pdf/GPT-1.pdf.

[15] Xu Han, Zhengyan Zhang, Ning Ding, Gu Yuxian, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al., Pre-trained models: past, present and future, AI Open 2 (2021) 225–250, https://doi.org/10.1016/j.aiopen.2021.08.002.

[16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv:1609.08144, 2016, https://doi.org/10.48550/arXiv.1609.08144.

[17] Karl Weiss, Taghi M. Khoshgoftaar, DingDing Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 1–40. https://journalofbigdata.springeropen.com/articles/10.1186/s40537016-0043-6.

[18] Yu-Cheng Zhou, Zhe Zheng, Jia-Rui Lin, Lu. Xin-Zheng, Integrating nlp and context-free grammar for complex rule interpretation towards automated compliance checking, Comput. Ind. 142 (2022) 103746, https://doi.org/10.1016/j.compind.2022.103746.

[19] Zhe Zheng, Ke-Yin Chen, Xin-Yu Cao, Xin-Zheng Lu, Jia-Rui Lin, Llm-funcmapper: Function identification for interpreting complex clauses in building codes via llm, arXiv preprint arXiv:2308.08728, 2023, https://doi.org/10.48550/arXiv.2308.08728.

[20] Zhe Zheng, Yu-Cheng Zhou, Ke-Yin Chen, Lu Xin-Zheng, Zhong-Tian She, Jia-Rui Lin, A text classification-based approach for evaluating and enhancing the machine interpretability of building codes, Eng. Appl. Artif. Intell. 127 (2024) 107207, https://doi.org/10.1016/j.engappai.2023.107207.

[21] Mohammed Al Qady, Amr Kandil, Concept relation extraction from construction documents using natural language processing, J. Constr. Eng. Manag. 136 (3) (2010) 294–302. https://ascelibrary.org/doi/abs/10.1061/(ASCE)CO.1943-7862.0000131.

[22] Jiansong Zhang, Nora M. El-Gohary, Extending building information models semiautomatically using semantic natural language processing techniques, J. Comput. Civ. Eng. 30 (5) (2016). C4016004, https://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.19435487.0000536.

[23] Xu Na, Ling Ma, Li Wang, Yongliang Deng, Guodong Ni, Extracting domain knowledge elements of construction safety management: rule-based approach using chinese natural language processing, J. Manag. Eng. 37 (2) (2021) 04021001. https://ascelibrary.org/doi/abs/10.1061/(ASCE)ME.1943-5479.0000870.

[24] Kaijian Liu, Nora El-Gohary, Similarity-based dependency parsing for extracting dependency relations from bridge inspection reports, Comput. Civil Eng. 2017 (2017) 316–323. https://ascelibrary.org/doi/abs/10.1061/9780784480823.038.

[25] Tianshu Li, Mohamad Alipour, Devin K. Harris, Context-aware sequence labeling for condition information extraction from historical bridge inspection reports, Adv. Eng. Inform. 49 (2021) 101333, https://doi.org/10.1016/j.aei.2021.101333.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L ukasz Kaiser, and Illia Polosukhin., Attention is all you need, Adv. Neural Inf. Proces. Syst. 30 (2017). https://pdf-reader-dkraft.s3.us-east-2.amazonaws.com/1706.03762.pdf.

[27] Yoon Kim, Carl Denton, Luong Hoang, Alexander M. Rush, Structured attention networks, arXiv preprint arXiv:1702.00887, 2017, https://arxiv.org/abs/1702.00887.

[28] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, Shahrokh Valaee, Recent advances in recurrent neural networks, arXiv preprint arXiv:1801.01078, 2017, https://doi.org/10.48550/arXiv.1801.01078.

[29] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, Guangjian Tian, Do rnn and lstm have long memory?, in: International Conference on Machine Learning PMLR, 2020, pp. 11365–11375, in: https://proceedings.mlr.press/v119/zhao20c.html.

[30] Yinhan Liu, Myle Ott, Naman Goyal, Du Jingfei, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692, 2019, https://doi.org/10.48550/arXiv.1907.11692.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, Jeff Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inf. Proces. Syst. 26 (2013), in: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901bAbstract.html.

[32] Kenneth Ward Church, Word2vec, Nat. Lang. Eng. 23 (1) (2017) 155–162. https://www.cambridge.org/core/journals/natural-languageengineering/article/word2vec/B84AE4446BD47F48847B4904F0B36E0B.

[33] Jeffrey Pennington, Richard Socher, Christopher D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543. https://aclanthology.org/D14-1162.pdf.

[34] Xiong Luo, Wenwen Zhou, Weiping Wang, Yueqin Zhu, Jing Deng, Attention-based relation extraction with bidirectional gated recurrent unit and highway network in the analysis of geological data, IEEE Access 6 (2017) 5705–5715. https://ieeexplore.ieee.org/abstract/document/8240917.

[35] Ji Zhu, Trevor Hastie, Kernel logistic regression and the import vector machine, J. Comput. Graph. Stat. 14 (1) (2005) 185–205, https://doi.org/10.1198/106186005X25619.

[36] Qing Ke, Comparing scientific and technological impact of biomedical research, J. Inf. Secur. 12 (3) (2018) 706–717, https://doi.org/10.1016/j.joi.2018.06.010.

[37] Gali Halevi, Henk Moed, Judit Bar-Ilan, Suitability of google scholar as a source of scientific information and as a source of data for scientific evaluation—review of the literature, J. Inf. Secur. 11 (3) (2017) 823–834, https://doi.org/10.1016/j.joi.2017.06.005.

[38] Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al., On the effect of pretraining corpora on in-context learning by a large-scale language model, arXiv preprint arXiv:2204.13509, 2022, https://doi.org/10.48550/arXiv.2204.13509.

[39] Swati P. Patil, B.V. Pawar, Removing non-relevant links from top search results using feature score computation, Bull. Pure Appl. Sci. Math. Stat. 37 (2) (2018) 311–320. https://bpasjournals.com/admin/upload/dynamic2/8-BPAS-E-86-2018.

[40] Yunyao Li, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan, H.V. Jagadish, Regular expression learning for information extraction, in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 2008, pp. 21–30. https://aclanthology.org/D08-1003.pdf.

[41] TRCA, 2021 erosion control infrastructures - asset management plan, 2021. https://trca.ca/.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[43] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450, 2016, https://doi.org/10.48550/arXiv.1607.06450.

[44] Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, Francesco Orabona, Understanding adamw through proximal methods and scale-freeness, arXiv preprint arXiv:2202.00089, 2022, https://doi.org/10.48550/arXiv.2202.00089.