ARCE: Augmented RoBERTa with Contextualized Elucidations for NER in Automated Rule Checking

Jian Chen*,†,#, Jiabao Dou‡,#
*Ningxia Transport Science Research Institute Co., Ltd., Yinchuan, China
†Ningxia Highway Engineering Quality Inspection Center (Co., Ltd.), Yinchuan, China
‡Department of Computer Science, Hong Kong Baptist University, Hong Kong

***Abstract*—Accurate information extraction from specialized texts is a critical challenge for automated rule checking (ARC) in the architecture, engineering, and construction (AEC) domain. While large language models (LLMs) possess strong reasoning capabilities, their deployment in resource-constrained AEC environments is often impractical. Conversely, standard efficient models struggle with the significant domain gap. Although this gap can be mitigated by pre-training on large, human-curated corpora, such approaches are labor-intensive and costly. To address this, we propose ARCE (Augmented RoBERTa with Contextualized Elucidations), a novel knowledge distillation framework that leverages LLMs to synthesize a task-oriented corpus, termed *Cote*, for incrementally pre-training smaller models. ARCE systematically explores the optimal strategy for knowledge transfer. Our extensive experiments demonstrate that ARCE establishes a new state-of-the-art on a benchmark AEC dataset, achieving a Macro-F1 score of 77.20% and outperforming both domain-specific baselines and fine-tuned LLMs. Crucially, our study reveals a less is more principle: simple, direct explanations prove significantly more effective for domain adaptation than complex, role-based rationales in the NER task, which tend to introduce semantic noise.**

***Index Terms*—Automated Rule Checking, Large Language Models, Named Entity Recognition.**

I. INTRODUCTION

The architecture, engineering, and construction (AEC) industry, a cornerstone of the global economy, operates under a complex framework of safety codes and regulatory requirements documented in vast volumes of unstructured text [1]–[3]. To navigate this complexity and mitigate human error, automated rule checking (ARC) has emerged as a vital technology for ensuring regulatory compliance [4], [5]. A critical bottleneck in developing effective ARC systems, however, lies in the rule interpretation stage. This stage necessitates the accurate extraction of semantic information from specialized texts—a task formally known as named entity recognition (NER) [6], [7].

Early attempts to address the NER challenge in the AEC domain primarily involved fine-tuning pre-trained language models (PLMs) such as BERT [8] and RoBERTa [9] on small, in-domain labeled datasets. While these models have achieved state-of-the-art (SOTA) performance in general domains, their effectiveness in specialized fields like AEC is often constrained by a significant domain gap [10], [11]. The specialized lexicon and complex syntactic structures found in regulation documents differ drastically from the general-purpose corpora used for the models' initial training. This mismatch limits the models' ability to capture domain-specific semantic dependencies, making robust information extraction a persistent challenge.

To bridge this domain gap, the research community has largely converged on two primary paradigms. The first is domain-adaptive pre-training (DAPT), exemplified by approaches like ARCBERT [2]. This method involves further pre-training general PLMs on massive, human-curated domain-specific corpora to inject domain knowledge [12]. While effective, DAPT is exceptionally labor-intensive and costly. Collecting and cleaning gigabytes of high-quality AEC texts requires significant domain expertise and engineering effort, making it difficult to scale or update as regulations evolve [13].

The second, more recent paradigm leverages the emergent capabilities of large language models (LLMs). Models such as GPT-5 [14] and Qwen3 [15] possess strong reasoning and zero-shot extraction capabilities. However, deploying billion-parameter LLMs directly for ARC tasks presents severe practical hurdles. In real-world AEC scenarios—often involving on-site devices or high-throughput document processing—the exorbitant computational cost, high inference latency, and memory footprint of LLMs are prohibitive.

Consequently, a new research frontier has opened: how to leverage the knowledge of LLMs to augment smaller, more efficient models [16]. This is often achieved through generative data augmentation [17], [18], where LLMs are used to synthesize labeled data or auxiliary knowledge to train smaller models [19], [20]. However, a critical scientific question remains unexplored in this context: What is the optimal format of knowledge for transfer? While the community has heavily focused on Chain-of-Thought (CoT) prompting to elicit complex reasoning from LLMs [21], [22], it is unclear whether such verbose and complex rationales are beneficial or detrimental when distilling knowledge into a smaller model with limited capacity for the NER task [23].

In this work, we address these challenges by proposing ARCE (Augmented RoBERTa with Contextualized Elucidations). Unlike traditional DAPT methods that rely on raw text, ARCE introduces a novel knowledge distillation framework that leverages an LLM to generate a high-quality, task-oriented

#These authors contributed equally to this work. *Corresponding author: jchen.ai@foxmail.com.

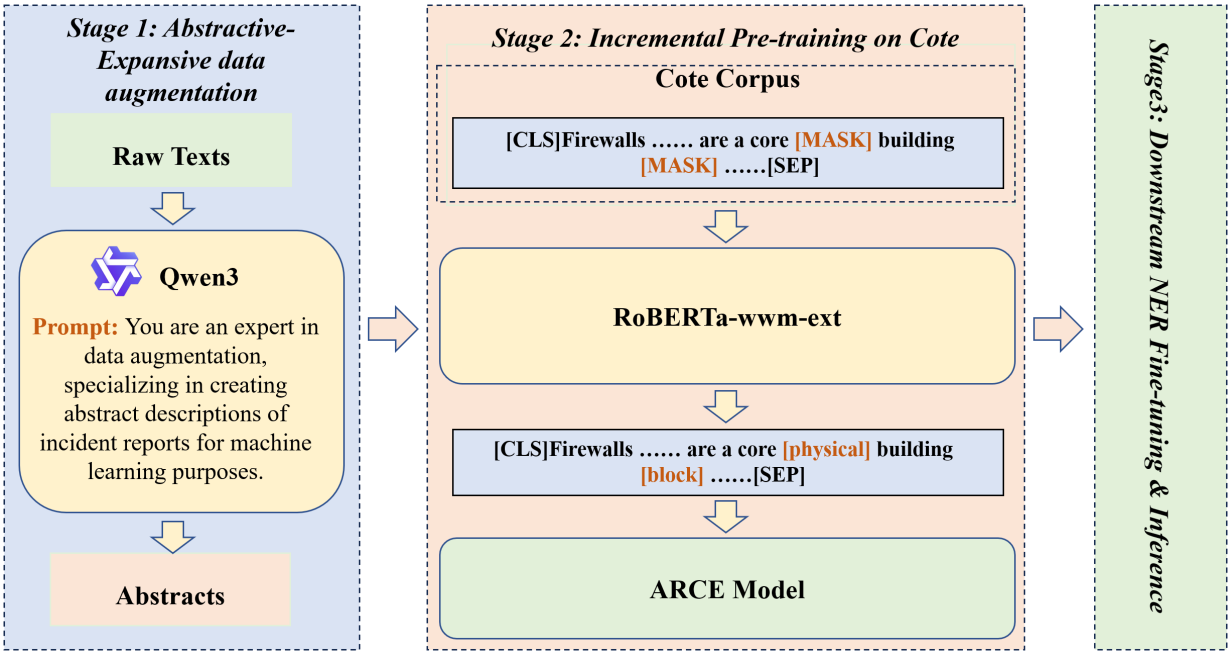

Fig. 1. The overall architecture of our proposed ARCE approach. The approach consists of three stages. **Stage 1**: We employ a LLM (e.g., Qwen3) to generate a Cote (contextualized task-oriented elucidation) corpus from raw domain texts using specialized prompts. **Stage 2**: A pre-trained RoBERTa-wwm-ext model undergoes incremental pre-training on the Cote corpus via a masked language modeling (MLM) objective, yielding the enhanced ARCE model. **Stage 3**: The ARCE model is augmented with a CRF layer and fine-tuned on the downstream NER task, and is then used for inference on new, unseen texts.

knowledge corpus—termed Cote. Instead of simply generating more training labels, ARCE tasks the LLM with providing contextualized elucidations—explicit semantic explanations of why a specific span constitutes an entity within its context. This corpus is then used to incrementally pre-train a RoBERTa model, effectively aligning its representation space with the domain-specific logic before fine-tuning.

Our approach offers a cost-effective alternative to massive pre-training while maintaining the inference efficiency of small language models (SLMs). Through extensive experiments, we validate that ARCE establishes a new state-of-the-art on a benchmark AEC dataset, outperforming both domain-adapted baselines and directly fine-tuned LLMs.

Crucially, our investigation yields a counter-intuitive and valuable insight regarding knowledge generation. We systematically compare different prompting strategies—ranging from complex, step-by-step reasoning to simple, direct explanations. We discover a principle: for domain adaptation of SLMs, a large volume of simple, explanation-based elucidations is significantly more effective than complex, role-based rationales. We hypothesize that while complex reasoning helps LLMs themselves, it introduces noise and overfitting risks for smaller models, whereas simple elucidations provide robust, linear semantic features that are easier to digest.

The main contributions of this paper are summarized as follows:

- We propose ARCE, a novel framework that leverages LLM-generated elucidations to effectively adapt a RoBERTa model to the AEC domain. This approach circumvents the high costs of collecting large human-curated corpora required by methods like ARCBERT and avoids the deployment latency of LLMs.
- We conduct a systematic ablation study of knowledge generation strategies. We reveal that simple, explanation-based elucidations significantly outperform complex, role-based reasoning for SLM adaptation, offering a guiding principle for future generative data augmentation research in the NER task.
- We establish a new state-of-the-art performance on the benchmark AEC NER dataset, surpassing strong baselines including domain-adapted BERT models and fine-tuned 8B-parameter LLMs, while maintaining a fraction of the inference cost.

II. METHODOLOGY

Our proposed approach, ARCE (augmented RoBERTa with contextualized elucidations), is designed to enhance NER performance in specialized domains by leveraging knowledge generated from a large language model, specifically Qwen3-8B [15]. As illustrated in Figure 1, the ARCE approach is decomposed into three sequential stages: (1) the generation of a contextualized task-oriented elucidation (Cote) corpus; (2) incremental pre-training of a RoBERTa model on this corpus; and (3) downstream NER fine-tuning with a conditional random field (CRF) layer.

A. Stage 1: Contextualized Task-Oriented Elucidation (Cote) Generation

The cornerstone of our approach is the creation of a high-quality, task-specific corpus, which we term the Corpus of contextualized task-oriented elucidations ($\mathcal{C}_{Cote}$). Unlike traditional data augmentation methods that rely on generic external corpora, we employ a powerful LLM, denoted as $\mathcal{M}$, to generate elucidations explicitly tailored to the reasoning process of the NER task.

Given a sentence $x = \{w_1, w_2, \ldots, w_n\}$ from a source dataset and an entity within it, defined by its text span s and entity type t, we construct a specialized prompt P based on a prompting strategy Π, such that $P = \Pi(x, s, t)$. This strategy directs the LLM to generate a textual elucidation k that clarifies why the span s corresponds to the type t within the context of x, or describes the functional role it plays.

The generation of the elucidation text $k = \{k_1, k_2, \ldots, k_L\}$ is an auto-regressive process, where the probability of the sequence is conditioned on the prompt P and the LLM's parameters $\theta_{\mathcal{M}}$. This is formally expressed as:

$$p(k|P; \theta_{\mathcal{M}}) = \prod_{i=1}^{L} p(k_i|k_{<i}, P; \theta_{\mathcal{M}}) \tag{1}$$

By iterating this process over all entities in our source data, we compile the final Cote corpus, $\mathcal{C}_{Cote}$, which serves as the foundation for adapting our NER model:

$$\mathcal{C}_{Cote} = \{k_j | k_j \sim p(k|\Pi(x_j, s_j, t_j); \theta_{\mathcal{M}})\} \tag{2}$$

where (x_j, s_j, t_j) represents the j-th entity instance in the source dataset.

B. Stage 2: Incremental Pre-training on Cote

To effectively integrate the domain-specific knowledge encapsulated in the elucidations into our NER model, we perform incremental pre-training on a RoBERTa model [24]. Let its initial parameters be $\theta_{RoBERTa}$. This stage adapts the general-domain model to the specific vocabulary, syntax, and reasoning patterns present in $\mathcal{C}_{Cote}$.

The training objective for this phase is the standard masked language modeling (MLM) task [8]. For each elucidation text $k \in \mathcal{C}_{Cote}$, we generate a corrupted version $\tilde{k}$ by randomly masking a subset of its tokens. Let k_{masked} be the set of original tokens at the masked positions. The MLM loss, $\mathcal{L}_{MLM}$, is defined as the negative log-likelihood of correctly predicting these original tokens from the corrupted context $\tilde{k}$:

$$\mathcal{L}_{MLM}(\theta) = -\sum_{k \in \mathcal{C}_{Cote}} \sum_{w \in k_{masked}} \log p(w|\tilde{k}; \theta) \tag{3}$$

The RoBERTa model's parameters are updated by minimizing this loss function over the entire Cote corpus. This procedure yields an enhanced set of parameters, θ_{ARCE}, which now encode both general linguistic understanding and the targeted task-specific knowledge, as shown in Eq. (4).

$$\theta_{ARCE} = \arg \min_{\theta_{RoBERTa}} \mathcal{L}_{MLM}(\theta_{RoBERTa}) \tag{4}$$

C. Stage 3: Downstream NER Fine-tuning with CRF

The final stage fine-tunes the adapted ARCE model for the downstream NER task. To effectively model dependencies between adjacent labels, we augment the architecture with a conditional random field (CRF) layer. The CRF layer computes a score $s(x, y) = \sum_{i=1}^{n} E_{i,y_i} + \sum_{i=1}^{n-1} A_{y_i,y_{i+1}}$ for a label sequence y, combining emission scores E and a learned transition matrix A. The model is jointly optimized by minimizing the negative log-likelihood of the ground-truth sequence y_{true}:

$$\mathcal{L}_{\text{NER}} = -\log \left(\frac{\exp(s(x, y_{\text{true}}))}{\sum_{y' \in \mathcal{Y}(x)} \exp(s(x, y'))} \right) \tag{5}$$

During inference, the Viterbi algorithm efficiently decodes the optimal label sequence $\hat{y} = \arg\max_{y' \in \mathcal{Y}(x)} s(x, y')$.

III. EXPERIMENTS

A. Experimental Setup

We evaluate our approach on the public AEC-domain NER dataset introduced by Zhou et al.[1] [6]. This dataset serves as a standard benchmark for regulatory compliance checking, containing 4,336 labeled entities distributed across 611 sentences drawn from diverse construction specifications. The entities cover specialized categories such as building components, constraints, and properties. Following the standard protocol established in previous works [2], we adopt an 8:1:1 split for training, validation, and testing. We report the Macro-F1 score as the primary metric to rigorously account for the label imbalance inherent in specialized domain corpora.

Our implementation leverages the PyTorch framework and the HuggingFace Transformers library. In the incremental pre-training stage, we train the base `RoBERTa-wwm-ext` model on our generated `Cote` corpus for 5 epochs. We utilize a learning rate of `5e-5`, a weight decay of `0.01` to prevent overfitting, and a batch size of 16. The masked language modeling (MLM) probability is set to 15%.

For the downstream NER fine-tuning, we train the augmented ARCE model for 10 epochs using the AdamW optimizer with a linear learning rate scheduler. To stabilize the training of the CRF layer, we employ a differential learning rate strategy: the CRF transition parameters are updated with a larger learning rate of `5e-1`, while the encoder parameters use a finer learning rate of `5e-5`. All experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU, demonstrating the computational efficiency of our approach compared to multi-GPU LLM training.

B. Baselines

To comprehensively assess the effectiveness of ARCE, we benchmark it against three distinct categories of baseline models, covering both traditional discriminative methods and cutting-edge generative paradigms:

1) **Traditional Fine-tuned Encoders:** We include standard pre-trained language models fine-tuned with a

[1]https://github.com/zhouyc98/auto-rule-transform

TABLE I
RESULTS COMPARISON WITH DIFFERENT METHODS. **BOLDED** VALUES INDICATE THE BEST PERFORMANCE.

Models	Macro-F1 with Strict Match (%)			Macro-F1 with Partial Match (%)		
	Precision	Recall	**Macro-F1**	Precision	Recall	**Macro-F1**
Traditional Methods						
NuNER-CRF [19]	64.61	62.79	63.69			
RoBERTa-CRF [9]	66.94	62.35	64.56		/	
BERT-base-Chinese-CRF [8]	77.54	71.02	72.93			
RoBERTa-wwm-ext-CRF [24]	74.62	68.36	71.19			
8B Large Language Models						
Llama3.1-8B-Instruct [25]	21.71	10.65	13.00	31.93	20.31	22,91
Ministral-8B-Instruct [26]	22.31	16.50	18.87	35.18	33.61	34.11
Qwen3-8B [15]	29.09	21.98	25.01	61.93	30.77	35.73
Large Language Models for Reasoning						
DS-R1-Distill-Llama-8B [22]	13.79	9.63	10.51	34.90	23.56	25.39
DS-R1-Distill-Qwen-7B [22]	19.26	10.00	12.47	31.86	17.41	21.23
DS-R1-0528-Qwen3-8B [22]	25.98	11.37	15.60	42.11	15.73	21.99
Qwen3-8B-think [15]	41.28	28.77	32.55	55.16	36.73	41.68
Qwen3-235B-A22B-Instruct-2507 [27]	29.98	28.14	28.65	71.98	46.08	49.13
Qwen3-8B-SFT [27]	75.48	75.39	75.23	84.07	83.41	**83.49**
ARCBERT-CRF [2]	77.31	75.26	75.75		/	
ARCE(Ours)	77.51	77.30	**77.20**		/	

CRF layer, including `BERT-base-Chinese` [8] and `RoBERTa-wwm-ext` [9]. These represent the widely used pre-train then fine-tune paradigm without domain adaptation.

2) **Large Language Models:** We evaluate both *Zero-Shot* and *Fine-Tuned (SFT)* performance of modern LLMs.
   - *General LLMs:* We test `Qwen3-8B` [15], `Llama3.1-8B` [25], and `Ministral-8B` [26] in a zero-shot setting to gauge their inherent domain understanding.
   - *Reasoning Models:* We include the recently proposed `DeepSeek-R1-Distill` series [22], which are optimized for complex reasoning chains, to investigate if heavy reasoning aids NER tasks.
   - *Fine-Tuned LLM:* We include `Qwen3-8B-SFT` [27], a model fine-tuned on the training set with LoRA [28], serving as a high-resource upper bound for generative approaches.
3) **Domain-Specific SOTA Methods:** We compare against `ARCBERT` [2], which uses a large human-curated domain corpus for pre-training, and `NuNER` [19], a state-of-the-art method employing LLM-annotated data for encoder pre-training.

C. Main Results and Analysis

The primary experimental results are presented in Table I. Our proposed ARCE approach establishes a new SOTA on the benchmark dataset, achieving a strict-match Macro-F1 score of 77.20%. ARCE significantly outperforms traditional methods, surpassing the base `RoBERTa-wwm-ext-CRF` by a substantial margin of 6.01 points. This improvement directly validates the efficacy of our `Cote` corpus in bridging the domain gap. Furthermore, ARCE surpasses the previous domain-specific SOTA, `ARCBERT` (75.26%), by roughly 2 points. Notably, ARCE achieves this without the need for the massive, human-curated corpora that `ARCBERT` relies on, proving the high value of synthetic, explanation-based knowledge.

A striking observation from Table I is the poor performance of zero-shot LLMs under the strict match criterion. While generative models possess strong semantic understanding, they struggle with precise boundary detection—often including punctuation or extra adjectives in the extracted span. When evaluated under a partial match metric, `Qwen3-8B`'s score improves to 35.73%, and the fine-tuned `Qwen3-8B-SFT` reaches a high 83.49%. However, in engineering contexts where strict compliance is required, extraction precision is non-negotiable. ARCE, being a token-level discriminative model, inherently avoids these boundary drift issues, offering a more reliable solution for automated rule checking.

Furthermore, we observe that reasoning models perform surprisingly poorly, scoring even lower than standard LLMs. Inspection of their outputs reveals that these models tend to over-think simple extraction tasks [29], often outputting verbose rationales mixed with the extraction results. This reinforces our finding that complex reasoning is not always beneficial for low-level structural tasks. Finally, comparing ARCE with `Qwen3-8B-SFT` (75.75%) highlights a critical efficiency advantage. While the fine-tuned 8B-parameter LLM achieves competitive results, it requires massive computational resources for inference. ARCE, based on the lightweight RoBERTa architecture, achieves a higher accuracy with less than 2% of the parameter count, making it uniquely suitable for deployment in resource-constrained AEC environments where running an 8B model is infeasible.

Strategy A: Explanation-Based

System Instruction:
You are a world-class requirements documentation analysis expert.

Task:
Please explain why '{span}' can be labeled as '{entity_type}' in the given text.

Constraint:
Your explanation must be logical, direct, and well-grounded in the context.

Applied Models:
- **ARCE:** Uses this prompt directly.
- **ARCE-think:** Activates thinking mode.

Strategy B: Role-Based

System Instruction:
You are a world-class requirements analysis expert. Your task is not simply to judge correctness, but to deeply analyze what **role** a fragment plays, what **functional effect** it has, and what **relationships** it has within the complete "text".

Reference Definitions:
- Target Type: {label_en}
- Definition: {label_def}

Analysis Task:
- Full Text: "{text}"
- Target Fragment: "{span}"

Constraint:
Deeply analyze the **role**, **function**, and **relationships**.

Fig. 2. Comparison of knowledge generation strategies. **Left (Strategy A):** The prompt used in our ARCE framework, focusing on simple explanations. **Right (Strategy B):** A complex prompt design forcing deep role analysis.

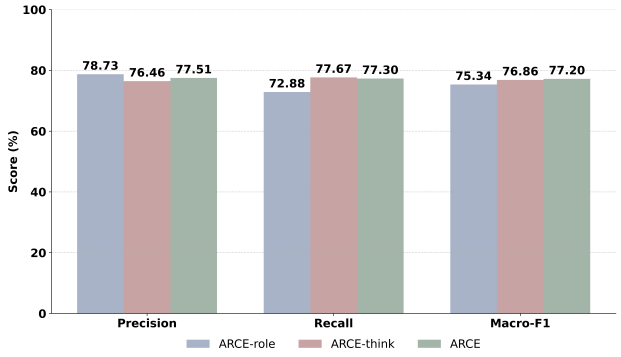

Fig. 3. Results of the ablation experiment.

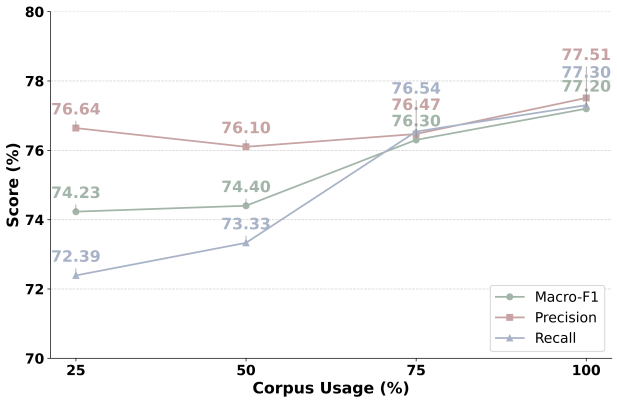

Fig. 4. Data Efficiency and Scalability Analysis.

IV. Discussion

A. Ablation on Knowledge Generation Strategy

To isolate the impact of the knowledge generation strategy and validate the less is more hypothesis, we conducted an ablation study comparing ARCE against two variants. ARCE utilizes *Strategy A* to generate simple, direct semantic explanations. In contrast, ARCE-think adds CoT reasoning traces to Strategy A, while ARCE-role employs *Strategy B* to analyze complex functional roles. As shown in Figure 2 and Figure 3, the standard ARCE achieves the highest Macro-F1 score, outperforming both variants.

A deeper analysis reveals that ARCE-role attains the highest precision but suffers from the lowest recall. We attribute this to a granularity mismatch. While *Strategy B* encourages intellectually rich, high-level abstractions, these descriptions do not map linearly to the token-level features required by the RoBERTa encoder. Consequently, the student model overfits to specific functional roles and fails to generalize to broader entity mentions.

Furthermore, ARCE-think unexpectedly underperforms the standard ARCE. This suggests that raw CoT traces act as semantic noise during distillation; the verbose intermediate reasoning steps dilute the core semantic signal, hindering the limited-capacity student model from extracting essential alignments. Collectively, these findings robustly support the principle that a large volume of simple elucidations provides a cleaner signal-to-noise ratio than complex reasoning chains for the domain adaptation of SLMs.

B. Data Efficiency and Scalability

To evaluate data efficiency and scalability, we pre-trained ARCE on increasing subsets (25%, 50%, 75%, and 100%) of the Cote corpus. As shown in Figure 4, performance exhibits a near-linear positive correlation with data volume. Remarkably, using just 25% of the corpus achieves a Macro-F1 of 74.23%, already surpassing the fully supervised RoBERTa-wwm-ext-CRF baseline by over 3 points. This highlights the high value density of LLM-generated elucidations, providing dense semantic signals that drastically reduce the data volume required for rapid domain alignment. Furthermore, the performance curve shows no signs of saturation typical of synthetic data methods. This sustained growth indicates that further gains are achievable simply by scaling the generation process, confirming ARCE's robust scalability as a data-centric solution for low-resource domains.

V. Conclusion

In this study, we proposed ARCE, a novel framework that adapts lightweight models to the AEC domain using LLM-generated explanations, establishing a new state-of-the-art with a 77.20% Macro-F1 score. Our investigation reveals a critical less is more principle: simple, direct elucidations facilitate superior knowledge transfer compared to complex reasoning chains in the NER task, which often introduce semantic noise for smaller models. By achieving high accuracy with minimal parameters, ARCE offers a viable solution for resource-constrained edge deployment. Future work will explore the generalizability of this paradigm across broader domains and its integration into end-to-end automated compliance checking systems.

Acknowledgment

This study was supported by the Transportation Power Special Fund Project of the Ningxia Hui Autonomous Region (No. 202500210) and the Ningxia Natural Science Foundation (No. 2026AAC020063).

References

[1] Aimi Sara Ismail, Kherun Nita Ali, and Noorminshah A Iahad. A review on bim-based automated code compliance checking system. In *2017 international conference on research and innovation in information systems (icriis)*, pages 1–6. IEEE, 2017.

[2] Zhe Zheng, Xin-Zheng Lu, Ke-Yin Chen, Yu-Cheng Zhou, and Jia-Rui Lin. Pretrained domain-specific language model for natural language processing tasks in the aec domain. *Computers in Industry*, 142:103733, 2022.

[3] Yunshun Zhong and Sebastian D Goodfellow. Domain-specific language models pre-trained on construction management systems corpora. *Automation in Construction*, 160:105316, 2024.

[4] Chengke Wu, Xiao Li, Yuanjun Guo, Jun Wang, Zengle Ren, Meng Wang, and Zhile Yang. Natural language processing for smart construction: Current status and future directions. *Automation in Construction*, 134:104059, 2022.

[5] Jaeyeol Song, Jinsung Kim, and Jin-Kook Lee. Nlp and deep learning-based analysis of building regulations to support automated rule checking system. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 35, pages 1–7. IAARC Publications, 2018.

[6] Yu-Cheng Zhou, Zhe Zheng, Jia-Rui Lin, and Xin-Zheng Lu. Integrating nlp and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry*, 142:103746, 2022.

[7] Jian Chen, Jinbao Tian, Ting Zheng, Yingxin Hui, and Zhou Li. Qwen-aec: Instruction-tuning large language models for high-performance building code analysis. *Advanced Engineering Informatics*, 71:104268, 2026.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

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

[10] Ruichuan Zhang and Nora El-Gohary. A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Automation in Construction*, 132:103834, 2021.

[11] Yu Gao, Xiaoxiao Xu, Tak Wing Yiu, and Jiayuan Wang. Transfer learning for smart construction: Advances and future directions. *Automation in Construction*, 175:106238, 2025.

[12] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[13] Zhe Zheng, Yu-Cheng Zhou, Ke-Yin Chen, Xin-Zheng Lu, Zhong-Tian She, and Jia-Rui Lin. A text classification-based approach for evaluating and enhancing the machine interpretability of building codes. *Engineering Applications of Artificial Intelligence*, 127:107207, 2024.

[14] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[15] Qwen Team. Qwen3 technical report, 2025.

[16] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, 2023.

[17] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*, 2022.

[18] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? gpt-3 can help. *arXiv preprint arXiv:2108.13487*, 2021.

[19] Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. Nuner: Entity recognition encoder pre-training via llm-annotated data. *arXiv preprint arXiv:2402.15343*, 2024.

[20] Zhihao Zhang, Sophia Yat Mei Lee, Junshuang Wu, Dong Zhang, Shoushan Li, Erik Cambria, and Guodong Zhou. Cross-domain ner with generated task-oriented knowledge: An empirical study from information density perspective. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1595–1609, 2024.

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[22] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[23] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

[24] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for chinese natural language processing. *arXiv preprint arXiv:2004.13922*, 2020.

[25] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[26] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[27] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*, 2024.

[28] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[29] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.