ENGinius: A Bilingual LLM Optimized for Plant Construction Engineering

Wooseong Lee[1], Minseo Kim[1], Taeil Hur[2], Gyeonghwan Jang[2], Woncheol Lee[2], Maro Na[2], Taeuk Kim[1]*

[1]Hanyang University [2]JENTI Inc.

{fokyoum, er1123090, kimtaeuk}@hanyang.ac.kr

{taei.hur, ghjang, woncheol, namaro825}@jenti.ai

Abstract

Recent advances in large language models (LLMs) have drawn attention for their potential to automate and optimize processes across various sectors. However, the adoption of LLMs in the *plant construction* industry remains limited, mainly due to its highly specialized nature and the lack of resources for domain-specific training and evaluation. In this work, we propose ENGinius, the first LLM designed for plant construction engineering. We present procedures for data construction and model training, along with the first benchmarks tailored to this underrepresented domain. We show that ENGinius delivers optimized responses to plant engineers by leveraging enriched domain knowledge. We also demonstrate its practical impact and use cases, such as technical document processing and multilingual communication.

1 Introduction

Recent progress in large language models (LLMs) has been driving innovation across diverse sectors. While general-purpose LLMs like ChatGPT (OpenAI, 2022) offer a solid foundation for various applications, complex and underexplored domains often require model adaptation to achieve behavior aligned with domain-specific requirements.

To this end, specialized LLMs have been developed for well-studied areas, e.g., healthcare (Zhang et al., 2023), finance (Wang et al., 2023), and law (Colombo et al., 2024). However, integrating LLMs into *plant construction engineering* (PCE) remains challenging, mainly due to the complexity of technical terms, the industry's multidisciplinary nature, and the lack of standardized domain-specific data.

In this study, we argue that, despite existing challenges, PCE is a high-priority sector that stands to benefit from the deployment of field-specific LLMs. Figure 1 and Table 1 provide intuitive evidence supporting the claim. Figure 1 illustrates the case that ChatGPT interprets the acronym 'NGS' as 'Next-Generation Sequencing,' a term widely recognized in life sciences (Schuster, 2008). However, in the context of PCE, it must be translated as 'Natural Gas System'. This implies that technical terms from PCE are likely underrepresented in corpora used to train LLMs, which may cause these models to struggle with PCE-related tasks.

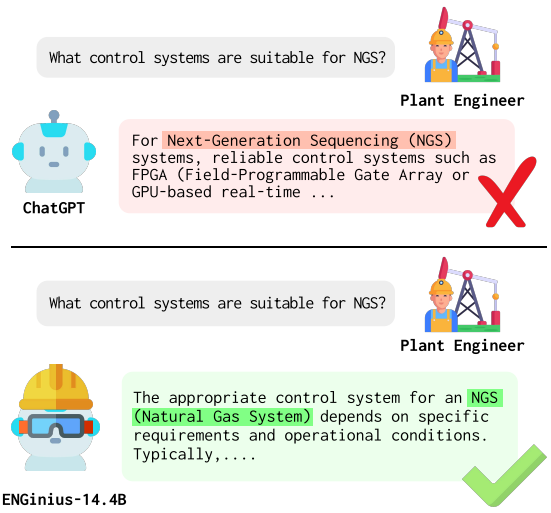

Figure 1: General LLMs (top) often struggle with domain-specific terminology and knowledge, frequently misinterpreting acronyms and specialized expressions. To overcome this challenge in the *plant construction* industry, we propose a novel LLM, ENGinius (below).

Furthermore, we emphasize that this issue is particularly acute for PCE, compared to other professional domains. Table 1 shows that while ChatGPT excels at understanding field-specific acronyms from the medical, financial, and legal disciplines, it largely fails to interpret PCE terms, even when provided with explanations of the target domain.[1] This

* Corresponding author

[1]We test ChatGPT's accuracy in explaining domain-specific acronyms, using 25 terms per domain. Each term is queried under two conditions: with and w/o domain info (i.e., name). The scores are averaged over 10 runs for robustness.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1350–1364

Domain	Success rates (%)	
	w/o domain info	w/ domain info
Medical	86.4%	100%
Finance	93.6%	100%
Law	60.0%	84.8%
PCE	**48.4%**	**55.6%**

Table 1: Comparison of ChatGPT's success rates in recognizing domain-specific acronyms with and without domain explanation. It falls well short in handling PCE.

result further highlights the limitations of general LLMs in handling unique domains, such as PCE.

In this work, we propose **ENGinius**, a novel LLM designed for the plant construction industry, to address the aforementioned challenges. The main contributions of this work are as follows:

1. As no suitable datasets currently exist, we first introduce a suite of **datasets** designed **for domain-adaptive pre-training & post-training in PCE**. ENGinius is trained on these new datasets, allowing it to be effectively optimized for the domain.

2. The problems caused by domain rarity can be more pronounced in multilingual settings. To investigate such issues, ENGinius is developed as a **bilingual model for English and Korean**.

3. Moreover, we propose two **novel benchmarks** to evaluate LLM performance **in realistic PCE scenarios**, part of which will be open-sourced. Experimental results on these new test sets show that ENGinius outperforms larger general-purpose LLMs in PCE-related tasks.

4. Finally, we showcase **real-world applications** implemented with ENGinius, e.g., expert and translation systems, highlighting its impact on improving work efficiency in the PCE domain.

2 Related Work

Interest in applying NLP to the PCE sector has been growing (Kim et al., 2018). Prior work has chiefly focused on technical document review—e.g., risky clause identification (Kim et al., 2022) and key contractual term extraction (Lee et al., 2020).

However, previous approaches to text processing in PCE have faced several limitations. The core problem stems from the scarcity and linguistic dissimilarity of the language used in PCE, which complicates the application of standardized rule-based (Winograd, 1972) and classification-based NLP techniques (Devlin et al., 2018). In addition, general NLP models (Young et al., 2018) are deficient in the specialized domain knowledge required in the PCE industry, often struggling to capture nuanced meanings embedded in complex contractual conditions, project dependencies, and implicit relationships between different document sections. This can lead to misinterpretation or incomplete analysis of PCE documents—e.g., misunderstanding key terms such as 'EOT' (Extension of Time) and 'LD' (Liquidated Damages).

Furthermore, the use of domain-specific language in multilingual or code-switching environments—which is common in companies outside English-centric countries—may exacerbate the aforementioned problems. To address these challenges, we propose ENGinius, a bilingual (English–Korean) language model tailored for PCE.

3 Benchmark Construction

A key prerequisite for effectively training and evaluating a domain-specific LLM is the establishment of a reliable benchmark within the target domain. Unfortunately, the PCE industry still lacks a suitable testbed for evaluating LLMs, partly due to its conservative and technically complex nature.

To alleviate this problem, we first introduce two novel *multiple-choice question (MCQ)* benchmarks dedicated to PCE: the **KOPIA** and **PE** benchmarks, targeting Korean and English, respectively. We aim to develop and validate a domain-specific LLM in bilingual settings, as data scarcity in specialized domains is often exacerbated by the additional complexity of multilingualism.

3.1 KOPIA Benchmark

We collaborate with the Korea Plant Industries Association (KOPIA)[2] to develop an industry-specific evaluation benchmark in Korean. This benchmark focuses on mechanical and piping engineering, a key subdomain of PCE, and covers terminology, technical standards, and process knowledge. Domain experts manually created and validated 1,000 test questions to ensure alignment with real-world practices. To support future research in the field, we plan to make this benchmark publicly available. See Appendix A.1 for more details.

[2]A government-affiliated organization that provides training for plant engineers (https://www.kopia.or.kr/).

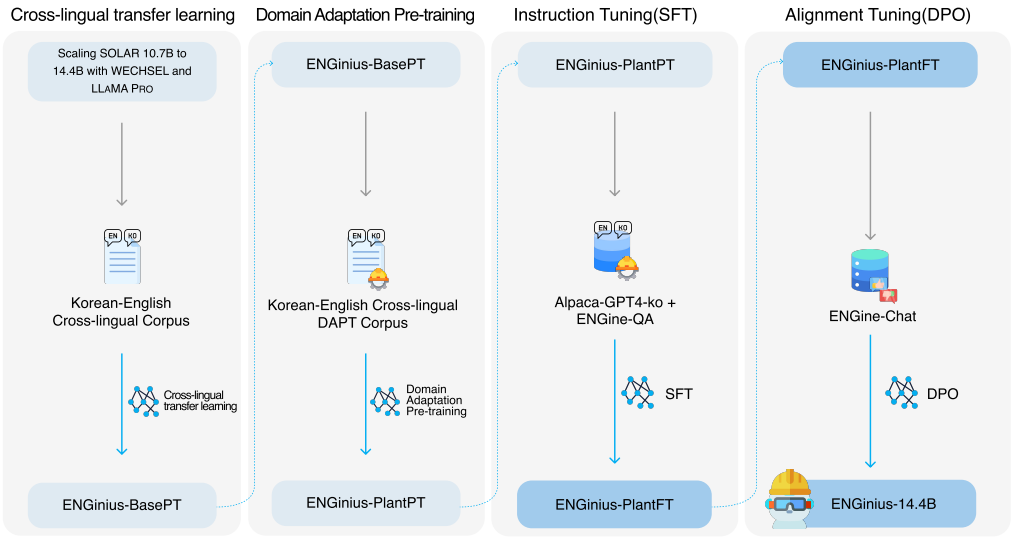

Figure 2: **Training procedure of ENGinius**. (1) SOLAR-10.7B is expanded to 14.4B using WECHSEL and LLaMA PRO, followed by bilingual training (**ENGinius-BasePT**). (2) Domain-Adaptive Pre-Training is then applied in the PCE domain, producing **ENGinius-PlantPT**. (3) The model is instruction-tuned to obtain **ENGinius-PlantFT**. (4) Finally, ENGinius-PlantFT is aligned via Direct Preference Optimization to produce the final **ENGinius-14.4B**.

3.2 Professional Engineer (PE) Benchmark

Inspired by MedQA US (Jin et al., 2020), we construct the Professional Engineer (PE) benchmark based on actual certification exams in the domain. It comprises 80 questions covering code knowledge, advanced calculations, and general conceptual understanding. This dataset is restricted to internal research use due to licensing constraints. Further details are provided in Appendix A.2.

4 Training of ENGinius

This section outlines the data collection and training procedures used to construct ENGinius. Since PCE is typically underrepresented in common textual resources, it is essential to first collect suitable industry-relevant corpora. We thus introduce a new suite of datasets developed for training ENGinius.

Furthermore, we detail the training procedure of ENGinius (see Figure 2), which leverages the corresponding datasets prepared for each stage. Table 10 provides exact configurations and hyperparameters. Each design choice is supported by extensive ablation studies reported alongside the training process.

4.1 Bilingual (English-Korean) Training

In the PCE industry, technical terms are often expressed in both English and a local language, requiring LLMs to possess strong bilingual capabilities. However, as existing LLMs are mostly trained on English-centric corpora (Grattafiori et al., 2024), they tend to exhibit suboptimal performance in relatively low-resource languages. (Ko et al., 2023).

To mitigate this issue, we selected SOLAR-10.7B (Kim et al., 2024) as our base model after evaluating several open-source alternatives (including Llama-2 13B (Touvron et al., 2023) and Mistral 7B (Jiang et al., 2023)). SOLAR-10.7B demonstrated strong performance on general language tasks and multilingual benchmarks, while offering the best balance between model size (10.7B parameters) and cross-lingual adaptability (see Table 11 in the Appendix for detailed ablation study results).[3]

Specifically, we employ the WECHSEL method (Minixhofer et al., 2022) to integrate new Korean tokens by initializing their embeddings using semantically similar English tokens. Subsequently, we adopt the LLaMA Pro methodology (Wu et al., 2024) to prevent catastrophic forgetting (Chen and Liu, 2018). Finally, we perform continued pretraining with an English-Korean bilingual corpus to induce cross-lingual transfer between the two languages, resulting in a new model named **ENGinius-BasePT**, which has 14.4B parameters.[4]

We verify the effectiveness of bilingual learning by comparing ENGinius-BasePT and SOLAR-

[3]The choice of language is guided by practical demand; however, in principle, our framework can be applied to any.

[4]See Appendix B for bilingual training and evaluation details. Note that the primary goal of this stage is to enhance the base model's general capabilities in English and Korean, rather than optimize it for a specific domain.

Datasets	Type	# of Tokens	Lang.
Plant Journals	Journal	7.75M	EN/KO
Civil, Architect	Books	89M	EN
Electric, Control, Safety	Books	145.3M	EN
Mechanical, Piping, HVAC	Books	173M	EN
Plant Commercial	Books	14.2M	EN/KO
Regulation & Standard Handbooks	Books	41.4M	EN/KO
National Competency Standards	Web Crawls	160.5M	KO
News	Web Crawls	1.52B	KO
Plant Papers	Paper	5.53B	EN/KO
Plant Articles	Article	8.87B	EN/KO
Total		**16.5B**	**EN/KO**

Table 2: Statistics of the datasets for Domain Adaptive Pre-Training (DAPT).

10.7B. As shown in Tables 8 and 9 in Appendix B, ENGinius-BasePT markedly outperforms the original on a Korean benchmark (Son et al., 2023) (78.09 vs. 59.57), while maintaining performance on English. This confirms that ENGinius-BasePT is well-suited as a foundation model for domain-specific training in the two target languages.

4.2 Domain-Adaptive Pre-Training (DAPT)

The multidisciplinary nature of PCE—covering mechanical, electrical, civil, architectural, and instrumentation disciplines—necessitates models that can comprehend diverse and interconnected domain knowledge. To cope with this complexity, we perform Domain-Adaptive Pre-Training (DAPT) (Gururangan et al., 2020) on ENGinius-BasePT, resulting in the domain-specialized model **ENGinius-PlantPT**, leveraging a wide range of PCE-related resources we collected (see Table 2).[5]

We compare ENGinius-BasePT and ENGinius-PlantPT to highlight the advantages of DAPT. The evaluation uses the KOPIA and PE benchmarks introduced in Section 3. As illustrated in Table 3, ENGinius-PlantPT consistently outperforms ENGinius-BasePT, underscoring the effectiveness of DAPT. We refer readers to Appendix C for the specifics of DAPT training and evaluation.

4.3 Instruction Tuning

In addition to DAPT, we explore domain-specific instruction tuning to further tailor the LLM for real-world applications. The goal of this phase is to adapt the model to more effectively handle tasks that align with the practical needs of stakeholders. To this end, our data suite—named **ENGine-QA** and summarized in Table 4—is designed to cover a range of practical tasks, including question answering, classification, dictionary prediction, and report generation. Note that this is manually constructed using a combination of in-house and open-source resources, the details of which are described below.

A core component of ENGine-QA is the Plant Expert QA subsets, derived from real-world discussions on ENG-TIPS, a globally recognized engineering forum.[6] By incorporating web-based comments and answers from domain experts into training, we expect the tuned model to naturally acquire specialized knowledge. We provide both English and Korean versions, with additional augmented data in Korean to improve bilingual coverage. Extra components in ENGine-QA are also included to provide effective training signals for the tuned model during instruction tuning. The role of each subset is described in detail in Appendix D.

On top of ENGine-QA, we also consider tuning the model with a general-purpose Korean instruction-following dataset to improve its language fluency and general reasoning ability. To this end, we translate the Alpaca-GPT4 dataset,[7] which contains diverse tasks generated by GPT-4 in a high-quality instruction–response format, and use it for instruction tuning. This dataset complements the domain-specific data (i.e., ENGine-QA) by enhancing general understanding and generation capabilities in Korean, which is particularly useful for tasks requiring broad linguistic competence.

To summarize, we produce **ENGinius-PlantFT** by instruction tuning using a combination of ENGine-QA and Alpaca-GPT4-ko, resulting in improved domain expertise, fluency, and language understanding. In the ablation study presented in Appendix D and Table 13, we demonstrate that our final configuration outperforms other feasible alternatives based on available resources.

4.4 Direct Preference Optimization (DPO)

Finally, we employ direct preference optimization (DPO) as the final step for training ENGinius. There is a risk that relying solely on instruction tuning with web-crawled datasets may degrade model quality, as user comments in forums such as ENG-TIPS are often noisy and imperfect. While some responses are grounded in industry standards, others may reflect subjective opinions or outdated practices To mitigate this issue and improve the relia-

[5] Before training, the domain-specificity of the datasets was validated through a visualization that highlights semantic gaps between our PCE datasets and general-purpose corpora. More details on this examination can be found in Appendix C.1.

[6] https://www.eng-tips.com/

[7] https://huggingface.co/datasets/llm-wizard/alpaca-gpt4-data/

Benchmark	KOPIA		PE		
Model	Pipe	Mech.	PE Calculation	PE Code	PE General
ENGinius-BasePT	44.85	50.61	29.41	**66.67**	38.71
ENGinius-PlantPT	**54.36**	**60.37**	**76.47**	**66.67**	**54.84**

Table 3: Performance before and after Domain-Adaptive Pre-Training (DAPT), evaluated on two benchmarks.

Components	Task	Quantity (EA)	Lang.
Plant Expert QA_KO case 1,2	QA	58,834	KO
Plant Expert QA_EN	QA	29,417	EN
Plant Discipline Classification	Classification	595	EN/KO
Plant Multiple Choice	MCQ	1,002	KO
Plant Terminology Dictionaries	Prediction	3,276	EN
Deviation Report	Generation	538	EN/KO
Total		**93,662**	**EN/KO**

Table 4: ENGine-QA components for instruction tuning.

Model	Mech.	Pipe	Avg.	Diff.
Gemma2-9B-it	58.64	59.39	57.89	-2.13 (-3.6%)
Orion-14B-Chat	51.96	52.32	51.61	-8.81 (-15.0%)
SOLAR 10.7B	50.65	53.13	48.17	-10.12 (-17.2%)
ENGinius 14.4B	**60.77**	**62.63**	**58.91**	-

Table 5: Performance comparison of the proposed model and baselines on KOPIA. **Diff.**: Diff from ENGinius.

bility of ENGinius, we apply DPO (Rafailov et al., 2023), a fine-tuning method that aligns model outputs with human or model-generated preferences.

To construct the DPO dataset, we again make use of Q&As from ENG-TIPS and generate two alternative responses per question using GPT-4o (OpenAI, 2024) and Mixture of Experts (MoE) prompting (Wang et al., 2024). All responses are generated in Korean. The specific steps for data construction and model training are as follows:

Two-Case Response Generation To capture variation in response quality and depth, we produce two distinct answers per question:

- **Case 1:** The original ENG-TIPS answer was anonymized and refined using GPT-4o for coherence and completeness.
- **Case 2:** MoE prompting generates a more context-rich and technically detailed response.

Human Preference Annotation Three senior specialists across mechanical, piping, electrical, and architectural disciplines evaluated response pairs and assigned preference scores based on predefined criteria (see Appendix E for more details). Responses were labeled as 'Chosen' or 'Rejected' based on aggregated scores.

Final Model Construction The generated dataset serves as the foundation for preference-based fine-tuning via DPO, resulting in the final **ENGinius-14.4B** model. This model is trained to generate responses aligned with expert expectations in real-world engineering contexts. To support research on domain-specialized LLMs, the DPO dataset will be publicly released.

5 Experimental Results

5.1 Experimental Settings

We adopt the LLM-as-a-judge framework (Zheng et al., 2023) to systematically evaluate model performance while minimizing human effort. For each question in the KOPIA and PE benchmark datasets, the tested models generate responses that are subsequently evaluated by LLaMA3-70B (Grattafiori et al., 2024), which serves as the judging model. Specifically, the judging model assesses correctness by comparing the generated responses with the provided reference solutions.

To ensure reliable and consistent evaluation, we conduct 20 independent runs for each model on the benchmarks. Final performance scores are computed by averaging the top five results from repeated evaluations.

5.2 Evaluation on the KOPIA Benchmark

Table 5 presents the experimental results of the proposed model and baseline methods on the KOPIA benchmark. Since all benchmark instances are multiple-choice questions, the reported scores represent the average accuracy over five runs. As baselines, we employ Gemma2-9B-it (Team et al., 2024), Orion-14B-Chat (Chen et al., 2024a), and SOLAR-10.7B (Kim et al., 2023). Experiments with external API-based models are excluded due to licensing constraints at the time of evaluation.

ENGinius-14.4B achieves an average score of 62 on the benchmark, outperforming baselines by nearly 3%-11%. The KOPIA test comprises two categories—Piping and Mechanical Engineering—in both of which ENGinius-14.4B demonstrates

Model	PE Test Code	PE Test Cal	PE Test General	Average	Diff. from ENGinius
Orion-14B-Chat	41.33	20.00	52.26	36.50	-31 (-45.9%)
GPT-3.5-turbo	60.00	47.06	45.16	48.75	-18.75 (-27.8%)
Gemma2-9B-it	72.00	34.71	59.99	51.50	-16 (-23.7%)
SOLAR 10.7B	72.00	40.59	54.83	52.00	-15.5 (-23.0%)
GPT-4	66.67	**52.94**	**74.84**	64.00	-3.5 (-5.2%)
ENGinius 14.4B (Ours)	**100**	46.47	**74.84**	**67.5**	-

Table 6: Performance comparison of the proposed ENGinius 14.4B and baselines, evaluated on the PE benchmark.

superior performance. These results confirm the model's effectiveness in understanding domain-specific knowledge essential to the PCE field.

5.3 Evaluation on the PE Benchmark

As in the previous subsection, the average accuracy of each model on the PE benchmark is reported in Table 6. The baselines include Orion-14B-Chat, GPT-3.5-turbo (OpenAI, 2023a), Gemma2-9B-it, SOLAR-10.7B, and GPT-4 (OpenAI, 2023b).

ENGinius-14.4B achieves an average score of 67.5, surpassing GPT-4's score of 64. Notably, while ENGinius-14.4B achieves higher average scores than GPT-4, our detailed analysis reveals important category-specific differences. GPT-4 demonstrates superior performance in the CAL[8] category, scoring 52.94 compared to ENGinius-14.4B's 46.47. This advantage likely stems from GPT-4's sophisticated mathematical reasoning capabilities, which benefit computation-intensive engineering questions.

While the Professional Engineer (PE) exam does not specify an official passing score, a score of approximately 65 is generally regarded as the passing threshold (NCEES, 2022). Accordingly, ENGinius-14.4B demonstrates superior performance over widely used proprietary models and open-source LLMs, meeting the level typically associated with certification-level expertise.

6 Real-World Applications

While we propose ENGinius as the first known application of a bilingual LLM in the PCE industry, we also share insights from its deployment. ENGinius is now actively utilized by a major company as the core of various real-world applications across different PCE workflows. Figures 3 and 5 (in Appendix F) illustrate a few representative cases.

Expert System As shown in Figure 3, ENGinius assists engineers by providing accurate answers

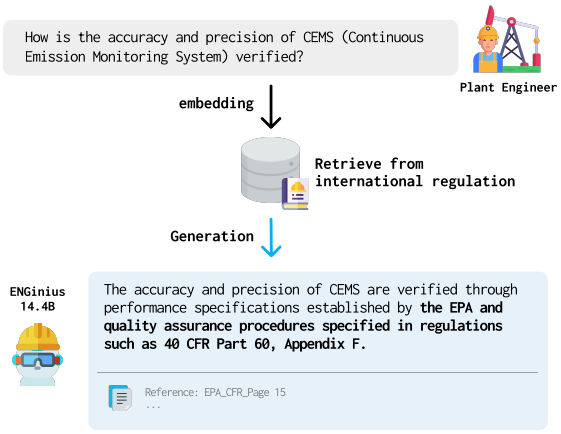

Figure 3: We share case studies of deploying ENGinius in an actual PCE industry environment. In this example, ENGinius functions as an expert system by retrieving accurate domain-specific knowledge and generating reliable responses aligned with engineering standards.

to technical questions. In addition, by utilizing Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), the system references internal design standards and technical codes to generate informed recommendations on engineering implementations.

Automated Document Analysis Given the complexity of Invitations to Bid (ITB) documents, manual review is inefficient and prone to error. ENGinius streamlines this process through contract risk assessment—retrieving semantically similar clauses from historical data—and change detection, which compares current and past terms to identify shifts in client requirements.

Client Letter & Deviation Report Generation Drafting of official project correspondence is another application. The model refers to previously approved documents and generates a draft that aligns with the current project's standards.

Document Translation PCE documentation often spans multiple languages, posing challenges for cross-lingual understanding. ENGinius lever-

[8]Calculation. See Appendix A-2 for details.

ages translation abilities, especially with handling cross-lingual PCE terminology.

7 Conclusion

In this work, we present ENGinius, the first LLM tailored for the plant construction engineering (PCE) domain. We construct bilingual training corpora and introduced two new benchmarks—KOPIA and PE—designed to evaluate model performance in realistic PCE scenarios. Through DAPT, instruction tuning, and DPO, ENGinius significantly outperforms general-purpose LLMs on PCE-specific tasks. Furthermore, its deployment in an industrial setting demonstrates tangible benefits across engineering workflows. Our research highlights the importance of domain-specialized LLMs in high-priority, yet underrepresented industries, and hope this work provides a foundation for further research in industrial NLP applications.

8 Future Work

8.1 Multilingual Expansion

While the current implementation of ENGinius focuses on Korean-English bilingual capabilities, the PCE industry is inherently international. Engineering specifications, contractual requirements, and technical standards frequently appear in multiple languages, depending on project locations and stakeholder nationalities.

Building upon our bilingual foundation, we aim to extend ENGinius into a multilingual framework capable of processing technical content across diverse languages. This will involve:

- Developing parallel corpora for low-resource technical languages;
- Exploring cross-lingual transfer methods tailored to engineering terminology;
- Handling inconsistencies in multilingual representations of technical concepts.

Such multilingual capabilities would significantly enhance ENGinius's utility in global engineering contexts, promoting better communication and knowledge sharing across international teams.

8.2 Retrieval-Augmented Generation Integration

We also plan to incorporate Retrieval-Augmented Generation (RAG) into ENGinius. Given the volume and complexity of PCE documentation, RAG can support more accurate retrieval and generation by:

- Constructing vector databases from domain-specific engineering codes and standards;
- Designing retrieval strategies tailored to technical language and hierarchical documentation structures; and
- Evaluating performance improvements in tasks such as design validation and compliance Q&A.

This integration would strengthen ENGinius's role as a practical tool for real-world engineering applications, bridging theoretical advancements with industrial utility.

Limitations

Data Constraints. In the PCE industry, authoritative information is primarily derived from international codes, which are copyrighted by various professional associations. This posed challenges in collecting and utilizing data for research purposes. Currently, some associations provide subscription-based text search services, but these are limited to keyword searches and do not support semantic search, making it difficult to extract relevant information effectively. In the future, if these constraints are addressed—particularly with the introduction of vector database-powered subscription services—API integration could enable more efficient data access and retrieval.

Computational Resource Limitations. The ENGinius model developed in this study is a large-scale language model (LLM) with approximately 14.4B parameters, requiring extensive GPU resources and significant training time. Although we initially constructed a dataset consisting of 388B English tokens and 194B Korean tokens, due to resource constraints, we could only train on 4.2B English tokens and 42.2B Korean tokens. Future improvements in computational resources would allow for the development of an even more powerful model.

Benchmark Limitations. The benchmarks introduced in this study were developed based on research-driven evaluation criteria. However, actual industry users may have different priorities, and the evaluation criteria used in this study may not fully align with real-world user experiences. Specifically,

field engineers' requirements, emergency response needs, and business-specific usage patterns might not be fully captured by our benchmarks. Therefore, we acknowledge that our benchmarks may not perfectly reflect real-world applications, and future research should incorporate user-based evaluations and feedback to enhance practical relevance.

Absence of RAG Evaluation.

This study focused primarily on the development and intrinsic performance evaluation of ENGinius, the first large-scale language model tailored for the Plant Construction Engineering (PCE) domain. Consequently, benchmark experiments involving Retrieval-Augmented Generation (RAG) were excluded from the current research scope. Nonetheless, RAG is a crucial technology for constructing document retrieval and question-answering systems in real-world industrial contexts. As discussed in Section 6.

Ethics Statement

The ENGinius model presented in this study is a large language model specialized for the plant construction industry, demonstrating how generative AI can be applied safely in this domain. To prevent the generation of offensive or harmful content, we implement ethical guardrails using DPO (Direct Preference Optimization) techniques. This involves filtering harmful content based on datasets such as Huggingface's `MrBananaHuman/kor_ethical_question_answer`, ensuring that the model adheres to ethical standards.

Furthermore, personal and sensitive information was rigorously removed during data preprocessing to ensure that the model meets ethical guidelines. Ethical considerations were also integrated throughout the training and evaluation processes, ensuring that the model remains safe and fair for application in real-world PCE industry settings.

Future research will not only focus on improving model performance but also on addressing diverse ethical issues, ultimately contributing to the development of a more reliable AI system.

Acknowledgments

This work was supported by the Technology development Program(RS-2024-00510893) funded by the Ministry of SMEs and Startups(MSS, Korea). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the artificial intelligence semiconductor support program to nurture the best talents (IITP-(2025)-RS-2023-00253914) grant funded by the Korea government(MSIT). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University))

References

American Petroleum Institute. Api standards online store. Accessed: 2025-03-22.

Du Chen, Yi Huang, Xiaopu Li, Yongqiang Li, Yongqiang Liu, Haihui Pan, Leichao Xu, Dacheng Zhang, Zhipeng Zhang, and Kun Han. 2024a. Orion-14b: Open-source multilingual large language models. *arXiv preprint arXiv:2401.12246*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Zhiyuan Chen and Bing Liu. 2018. *Continual Learning and Catastrophic Forgetting*, pages 55–75. Springer International Publishing, Cham.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. Saullm-7b: A pioneering large language model for law. *Preprint*, arXiv:2403.03883.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Myeongjun Jang, Dohyung Kim, Deuk Sin Kwon, and Eric Davis. 2022. KoBEST: Korean balanced evaluation of significant tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3697–3708, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Preprint*, arXiv:2009.13081.

I. T. Jolliffe. 2016. Principal component analysis. *Springer Series in Statistics*.

Chae-Yeon Kim, Jong-Gwan Jeong, So-Won Choi, and Eul-Bum Lee. 2022. An ai-based automatic risks detection solution for plant owner's technical requirements in equipment purchase order. *Sustainability*, 14(16):10010.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, and 1 others. 2023. Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. *arXiv preprint arXiv:2312.15166*.

J. Kim, S. Park, and H. Lee. 2018. Extraction of critical contract terms from construction contracts using natural language processing techniques. In *Proceedings of the ASCE International Conference on Construction Engineering*.

Sanghoon Kim, Dahyun Kim, Chanjun Park, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2024. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 23–35, Mexico City, Mexico. Association for Computational Linguistics.

Hyunwoong Ko, Kichang Yang, Minho Ryu, Taekyoon Choi, Seungmu Yang, Jiwung Hyun, Sungho Park, and Kyubyong Park. 2023. A technical report for polyglot-ko: Open-source large-scale korean language models. *Preprint*, arXiv:2306.02254.

Eul-Bum Lee, Chae-Yeon Kim, Jong-Gwan Jeong, and So-Won Choi. 2020. Application of natural language processing (nlp) and text-mining of big-data to engineering-procurement-construction (epc) bid and contract documents. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5645–5654. IEEE.

Patrick Lewis, Ethan Perez, Aleksandru Constantin, and et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

National Fire Protection Association. 2022. *2023 National Electrical Safety Code*. Institute of Electrical and Electronics Engineers, Quincy, MA. Accessed: 2025-03-22.

National Fire Protection Association. 2023. *NFPA 70: National Electrical Code*, 2023 edition. National Fire Protection Association, Quincy, MA. Accessed: 2025-03-22.

NCEES. 2022. Professional engineering (pe) examination information.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023a. Gpt-3.5-turbo.

OpenAI. 2023b. Gpt-4 technical report.

OpenAI. 2024. Gpt-4o system card.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Stephan C Schuster. 2008. Next-generation sequencing transforms today's biology. *Nature methods*, 5(1):16–18.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jaecheol Lee, Je Won Yeom, Jihyu Jung, Jung Woo Kim, and Songseong Kim. 2023. Hae-rae bench: Evaluation of korean knowledge in language models. *arXiv preprint arXiv:2309.02706*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. FinGPT: Instruction tuning benchmark for open-source large language models in financial datasets. In *Workshop Instruction Tuning and Instruction Following @ NeurIPS 2023*. Accepted in Oct 2023.

Ruochen Wang, Sohyun An, Minhao Cheng, Tianyi Zhou, Sung Ju Hwang, and Cho-Jui Hsieh. 2024. One prompt is not enough: Automated construction of a mixture-of-expert prompts. *arXiv preprint arXiv:2407.00256*.

Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. LLaMA pro: Progressive LLaMA with block expansion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6518–6537, Bangkok, Thailand. Association for Computational Linguistics.

Tom Young, Diarmuid Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Xinlu Zhang, Chenxin Tian, Xianjun Yang, Lichang Chen, Zekun Li, and Linda Ruth Petzold. 2023. Alpacare:instruction-tuned large language models for medical application. *Preprint*, arXiv:2310.14558.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, and Eric P. Xing. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Details on Benchmark Construction

A.1 The KOPIA Benchmark

This dataset, created in partnership with KOPIA, evaluates key competencies in plant engineering across three dimensions:

- **Terminology:** Correct understanding and usage of industry-specific terms.
- **Technical Standards:** Interpretation and application of engineering codes and industry specifications.
- **Process Knowledge:** Understanding workflows, procedures, and problem-solving in EPC projects.

Development Process

- KOPIA coordinated industry experts to develop 500 mechanical and 500 piping engineering questions (total 1,000).
- Our research team provided technical oversight, with final validation conducted by Professional Engineers (PEs).

A.2 The Professional Engineer (PE) Benchmark

Inspired by established domain-specific evaluation datasets (e.g., MedQA US, MedMCQA(Pal et al., 2022)), we constructed the PE Exam-based dataset as follows:

- Publicly available PE exam-style questions were collected through web crawling and manual curation.
- The dataset mirrors official PE exam difficulty distributions and syllabus topics, emphasizing plant engineering and power systems.

Dataset Composition The dataset contains 80 questions categorized as:

- **Code Knowledge (15 questions):** API(American Petroleum Institute), NEC(National Fire Protection Association, 2023), NESC(National Fire Protection Association, 2022) standards.
- **Advanced Calculations (34 questions):** Technical problem-solving.
- **General Conceptual Knowledge (31 questions):** Foundational engineering concepts.

Dataset	Type	Training Data Volume (# of Tokens)
English Dataset	Book Web text ArXiv Github Etc.	4.2B
Korean Dataset	Web text Dictionary Report Corpus Data Etc.	42.2B
Total		**46.4B**

Table 7: A bilingual dataset for continued pretraining.

This dataset serves as a supplementary evaluation tool to gauge ENGinius's capability in solving complex technical tasks.

B Details on English-Korean Bilingual Learning and Evaluation

As shown in Table 7, the English-Korean bilingual dataset was constructed using a 10:1 ratio of Korean to English data. We assess the cross-lingual performance of ENGinius-BasePT by evaluating it separately on English and Korean benchmarks.

For English, the model was tested on widely used benchmarks including ARC(Clark et al., 2018) (scientific reasoning), GSM8K(Cobbe et al., 2021) (mathematical problem solving), HellaSwag(Zellers et al., 2019) (commonsense reasoning), MMLU(Hendrycks et al., 2021) (broad domain knowledge), TruthfulQA(Lin et al., 2022) (truthful reasoning), and Winogrande(Sakaguchi et al., 2021) (contextual understanding). As shown in Table 8, **ENGinius-BasePT** maintained competitive performance, with only a minor drop of 1.8% compared to **SOLAR-10.7B** (64.21 vs. 66.01), indicating effective mitigation of catastrophic forgetting.

For Korean, we used the Haerae benchmark (Son et al., 2023), which includes five categories: Loan Words (distinguishing refined Korean from borrowed terms), Standard Nomenclature (use of standardized professional terminology), Rare Words (understanding uncommon vocabulary), General Knowledge (cultural, legal, and entertainment knowledge), and History (factual understanding of Korean history). As shown in Table 9, **ENGinius-BasePT** significantly outperformed the baseline across all categories, achieving a total improvement of 18.5% (78.09 vs. 59.57), demonstrating the

Model	ARC Challenge	GSM8K	HellaSwag	MMLU	TruthfulQA(MC2)	Winogrande	Average
ENGinius-BasePT	61.01	48.82	84.00	63.37	45.61	82.48	64.21
SOLAR-10.7B	61.35	55.50	84.55	65.52	45.65	83.50	66.01

Table 8: Comparison of performance before and after bilingual training on various English benchmarks.

Model	Average	General Knowledge	History	Loan Word	Rare Word	Standard Nomenclature
ENGinius-BasePT	78.09	51.70	85.64	84.62	80.74	84.97
SOLAR-10.7B	59.57	39.77	54.78	69.23	63.70	66.66

Table 9: Comparison of performance before and after bilingual training on the Korean (Haerae) benchmark.

effectiveness of cross-lingual pretraining in enhancing Korean performance while preserving English capability.

Category	Details
DAPT (Full Finetuning)	
Learning Rate	$1.0e^{-5}$
Batch Size	1024
Context Length	4096
Instruction Tuning (LoRA)	
Learning Rate	$1.0e^{-4}$
Batch Size	128
Context Length	4096
LoRA r	16
LoRA α	16
LoRA Dropout	0.05
DPO (LoRA)	
Learning Rate	$5.0e^{-6}$
Batch Size	32
Context Length	4096
LoRA r	16
LoRA α	16
LoRA Dropout	0.05

Table 10: Training environment and hyperparameters for each training stage.

C Details on Domain Adaptive Pre-Training (DAPT)

Table 2 provides an overview of the sources used to construct the DAPT dataset. Each component was selected to ensure coverage of essential disciplines such as mechanical, piping, electrical, and civil engineering, as well as regulatory standards and procurement-related materials.

The DAPT dataset integrates diverse sources to reflect domain-specific language and knowledge in engineering. It includes **plant journals** (2018–2023) on technologies and trends in PCE fields; materials on **civil and architectural** engineering; and references aligned with IEC, IEEE, NFPA, and ISA **standards**. **Technical guidelines** based on API and ASME cover mechanical, piping, and HVAC systems. The dataset also includes **government data** on plant terminology, contracts, and procurement; Korea's National Competency Standards (NCS); curated **news articles** (2020–2023); regulatory **handbooks** from agencies like the U.S. EPA and OSHA; and **technical papers** from APIs such as ScienceON and DBPia. All data were preprocessed to remove redundancy, enhance clarity, and match real-world engineering language.

C.1 PCA-Based Semantic Analysis

To demonstrate that our DAPT dataset captures the nuances of domain-specific terminology and context, we conduct a toy experiment on comparing semantic characteristics between PCE-specific data with those of general-domain data. Using BGE-M3 embeddings (Chen et al., 2024b) and Principle Component Analysis (PCA) (Jolliffe, 2016), we show clear separation of semantic vectors between general and PCE-specific texts. This demonstrates that the dataset reflects meaningful domain-specific distinctions.

To validate the uniqueness of the DAPT dataset, we performed PCA on semantic embeddings generated using the BGE-M3 embedding model. We compared samples from general-domain corpora and our DAPT dataset.

As shown in Figure 4, the embeddings from domain-specific texts form clusters distinct from those of general texts. This indicates that terms commonly used in both domains (e.g., beam, load, valve) exhibit significantly different semantic contexts, justifying the need for domain-specialized training data.

We highlighted two example sentences containing the word beam to illustrate this difference:

- *"A concentrated **beam** of light was emitted from the laser pointer."*

Model	Average	kobest_boolq	kobest_copa	kobest_hellaswag	kobest_sentineg	kobest_wic
basePT_solar	0.784	0.896	0.801	0.576	0.718	0.668
basePT_llama	0.759	0.798	0.830	0.642	0.985	0.540
basePT_mistral	0.596	0.511	0.724	0.542	0.980	0.488

Table 11: Performance comparison of bilingual pretraining using the same corpus on different base models: Llama 2, Mistral, and SOLAR. All models were trained with the same bilingual dataset and evaluated on the Korean benchmark KoBEST (Jang et al., 2022).

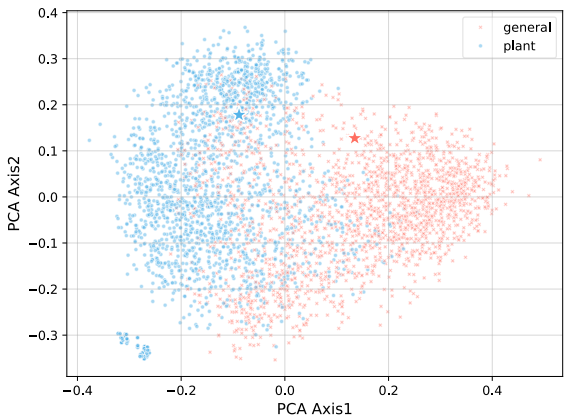

Figure 4: Embedding Distributions of General and Domain-Specific Data Visualized Using PCA.

Method	Pipe	Mech.	Average
Up-sampling	59.15	60.32	59.74
Down-sampling	62.63	58.91	60.77

Table 12: Performance comparison based on sampling strategy.

- *"The structural integrity of the steel **beam** must be verified to ensure compliance with ASCE design standards."*

These sentences are embedded in separate regions of the PCA space, supporting our claim that context-sensitive semantics are critical for industrial LLM performance.

C.2 Experiments on DAPT Sampling

Additional analyses and detailed results comparing sampling strategies (up-sampling vs. down-sampling).

Given the inherent imbalance among different data sources in the DAPT dataset, we compared two sampling strategies to improve domain-specific learning: Up-sampling and Down-sampling. Experiments evaluated using the KOPIA dataset revealed superior performance of down-sampling, especially notable in piping domain accuracy (improvement of 3.48%, detailed in Table 12). Therefore, down-sampling was adopted for subsequent experiments.

D Details on Instruction Tuning

The instruction tuning dataset was designed to enhance domain-specific reasoning (Chung et al., 2024), structured response generation, and terminology handling in the construction and plant industries. It includes data from diverse engineering disciplines, ensuring balanced representation. Below, we provide detailed descriptions of its key components.

Plant Expert QA The Plant Expert QA dataset, sourced from ENG-TIPS, captures real-world engineering discussions. It focuses on contextual term usage, helping the model accurately interpret engineering concepts in real scenarios.

To prevent domain bias, the dataset was structured to maintain balanced representation across mechanical, piping, electrical, instrumentation, civil, and architectural disciplines.

Classification This dataset enables the model to categorize technical documents and inquiries by discipline (e.g., mechanical, electrical, instrumentation). It improves the model's ability to identify and organize engineering content, supporting efficient information retrieval.

Deviation Report Generation Deviation reports document discrepancies between contract specifications and field conditions. This dataset trains the model to analyze deviations, generate structured reports, and ensure compliance with industry standards, aiding contract evaluation and project management.

Multiple Choice (MCQ) The MCQ dataset, designed to align with benchmark evaluations, includes questions on technical concepts, safety protocols, and regulatory standards. It enhances the model's precision in structured assessments.

Model	Mech.	Pipe	Avg.	Diff.
ENGinius-PlantPT	57.09	55.87	56.48	-
ENGinius-AG4FT	55.87	53.04	54.45	-2.0 (-3.6%)
ENGinius-KoPlantFT	61.13	58.70	59.92	+3.4 (+6.1%)
ENGinius-PlantFT	**63.77**	**60.45**	**62.11**	**+5.6 (+10.0%)**

Table 13: Performance of instruction-tuned model variants on the PE benchmark. **Diff.**: Difference from ENGinius-PlantPT.

Domain Dictionaries Engineers rely on domain-specific terminology and abbreviations. This dataset refines the model's understanding of frequently used technical terms, improving accuracy in document interpretation and engineering communication.

Alpaca-GPT4-ko In addition to domain-specific data, we also incorporated a general-purpose instruction-following dataset in Korean to improve the model's language fluency and general reasoning ability. For this, we translated and adapted the Alpaca-GPT4 dataset,[9] which contains diverse tasks generated by GPT-4 in a high-quality instruction-response format. This dataset complements the domain-specific data by enhancing general understanding and generation capability in Korean, especially useful for tasks requiring broad linguistic competence.

Ablation study for instruction tuning In this section, we conduct an ablation study to validate the effectiveness of each component used in instruction tuning. Below, we present the baseline models and our final model, **ENGinius-PlantFT**:

- **ENGinius-PlantPT:** The model only trained with DAPT.
- **ENGinius-AG4FT:** Fine-tuning ENGinius-PlantPT on Alpaca-GPT4-ko.
- **ENGinius-KoPlantFT:** Fine-tuning ENGinius-PlantPT with the combination of Alpaca-GPT4-ko and the Korean subset of ENGine-QA.
- **ENGinius-PlantFT:** Fine-tuning ENGinius-PlantPT with all instruction tuning data.

Table 13 shows that integrating both Alpaca-GPT4-ko and ENGine-QA yields the most significant improvement in domain expertise and linguistic quality.

[9] https://huggingface.co/datasets/llm-wizard/alpaca-gpt4-data/

E Details on Direct Preference Optimization (DPO)

E.1 DPO Evaluation Criteria

To ensure high-quality preference-based fine-tuning, domain experts evaluated response pairs using the following five criteria. Each response was rated on a 1–3 scale per criterion, with higher scores indicating stronger alignment with expert expectations.

- **Expertise** – Technical accuracy and adherence to verified engineering standards.
- **Clarity** – Clear and precise communication of key information.
- **Relevance** – Applicability of the response to the construction and plant engineering domain.
- **Conciseness** – Elimination of unnecessary details while preserving essential content.
- **Consistency** – Logical structure and coherence in addressing the question.

Based on the aggregated scores, responses were categorized as Chosen (preferred) or Rejected (non-preferred). These evaluations serve as the foundation for Direct Preference Optimization (DPO), enabling the model to prioritize expert-aligned responses in real-world engineering applications.

F Real-World Applications

In addition to the example in Table 3, Figure 5 provides examples of ENGinius in core engineering tasks.

ENGinius supports the generation of client letters and deviation reports by referencing past technical standards and previously approved documents. This allows engineers to produce consistent and contextually accurate drafts with minimal manual effort.

The model also enables automated analysis of document differences to identify changes in technical requirements, thereby improving the efficiency and reliability of contract review processes.

Finally, ENGinius handles translation of domain-specific content across languages, facilitating accurate and fluent cross-lingual understanding.

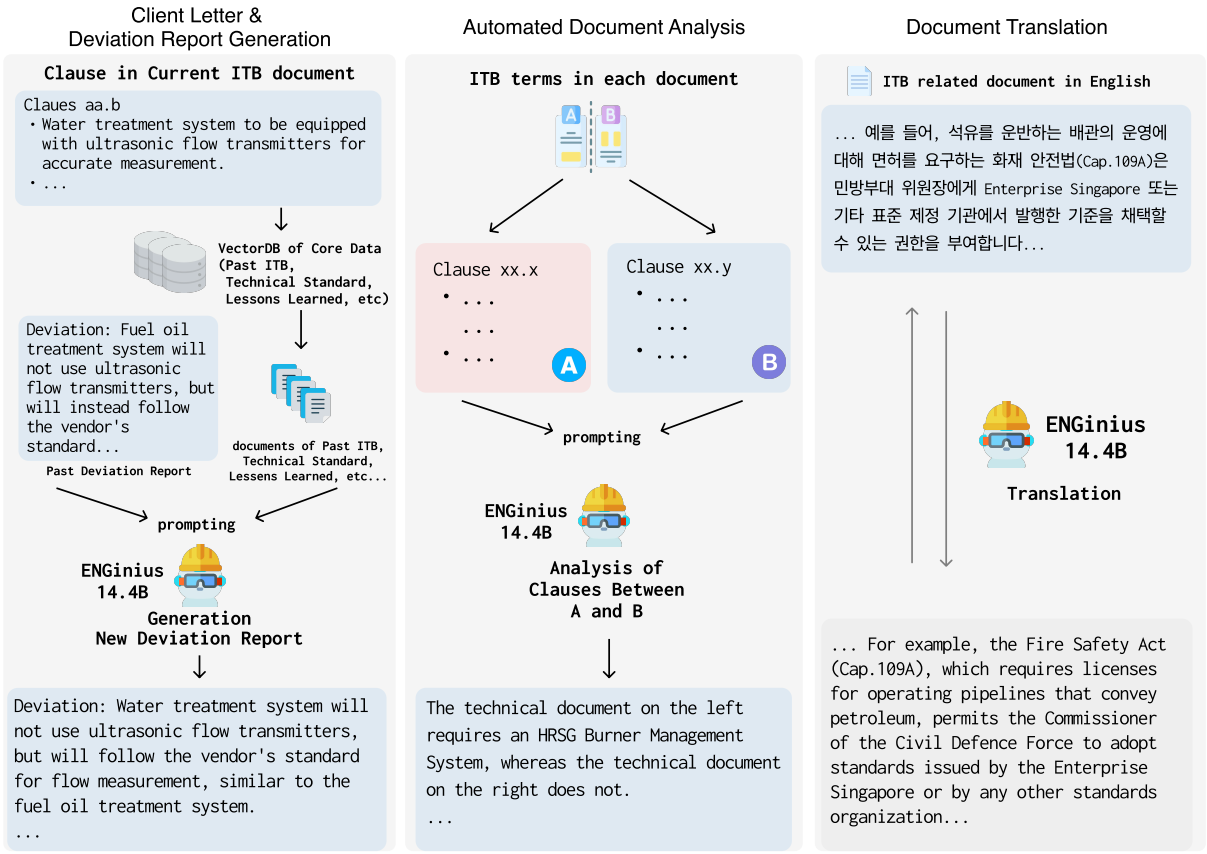

Figure 5: Real-world deployment of ENGinius across three core engineering tasks. Left: Generation of client letters and deviation reports by referencing prior documents. Center: Automated analysis of ITB documents to detect requirement changes. Right: High-fidelity translation of technical content to support multilingual understanding in engineering workflows.