Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

# Smart Prompt: Leveraging LLMs for Context-Aware Prompt Engineering Across Diverse Domains

[1]Rahul Sao, [2]Baibhava Patel, [3]Ashu Gupta, [4]Roshan Goswami, [5]Mr. Baidyanath Ram

[1,2,3,4]Students of Amity School of Engineering and Technology, [5]Assistant Professor

[1,2,3,4,5]Amity University Chhattisgarh

**Abstract -** Prompt engineering has become a crucial ability in the rapidly changing field of artificial intelligence for maximizing the capabilities of large language models (LLMs). The quality and structure of the input prompt continue to be crucial in influencing the relevance, depth, and inventiveness of the model's answer, even with the impressive capabilities of models like as GPT and Mistral. However, many users, particularly those without technical expertise, still find it difficult to create effective prompts across a variety of fields, including software development, picture generation, healthcare, legal analysis, and academic writing.

In this paper, a refined Mistral 7B model-based context-aware prompt augmentation system, Smart Prompt, is presented. Developed as an online tool, Smart Prompt lets users provide task descriptions in natural language, from which it automatically derives important parameters, target domain, and intent. Through the use of domain-specific training data and lightweight fine-tuning approaches, the system dynamically converts ambiguous or partial user inputs into highly impactful, well-structured prompts that are customized for certain LLM use cases. A straightforward request like "build a food delivery app" is improved, for instance, by recommending pertinent technologies, required parts, and contextual information that is lacking.

Smart Prompt automates a typically manual, expertise-driven procedure, bridging the gap between casual users and advanced prompt engineering. Our findings show enhanced job accuracy, better prompt quality, and more user satisfaction across a variety of fields. This work demonstrates a scalable strategy for democratizing prompt engineering through intelligent help, in addition to emphasizing the significance of carefully constructed prompts.

**Keywords:** Prompt Engineering, Large Language Models, Intent Extraction, Context-Aware AI, Domain-Specific Prompting, Natural Language Understanding

## 1. Introduction

Prompt engineering has surfaced as an indispensable skill for effective interaction with large language models, fundamentally shaping the model's responses through carefully crafted instructions [19,50]. This process involves designing and refining prompts to elicit the desired outputs from these models, thereby maximizing their utility across a multitude of applications [2,17]. As large language models become increasingly integrated into diverse sectors, the significance of prompt engineering in optimizing their performance and ensuring responsible application cannot be overstated [2].

Prompt engineering is not merely about formulating questions, but about understanding the nuances of language models and tailoring prompts to align with their inherent capabilities and limitations [2]. It requires a deep understanding of the task at hand, as well as the ability to anticipate how the model will interpret and respond to different phrasings [27].

The effectiveness of prompt engineering hinges on its ability to guide large language models towards established human logical thinking, surpassing the limitations of anthropomorphic assumptions that often underpin prompt design [27]. By incorporating key elements, well-constructed prompts enable large

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

language models to generate high-quality answers. Prompt engineering has evolved into a form of programming, enabling users to customize interactions with large language models, automate processes, and enforce specific qualities of generated output [50].

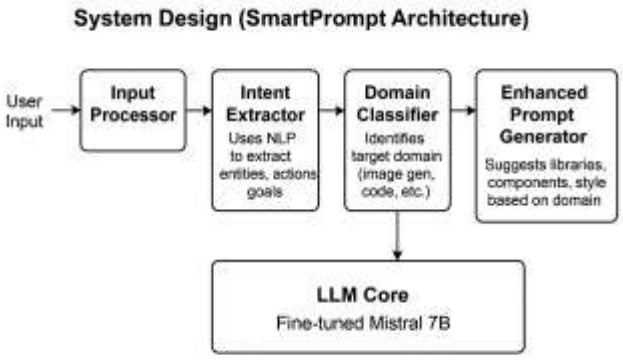
Large language models (LLMs) have rapidly transformed the landscape of artificial intelligence, demonstrating remarkable capabilities in tasks such as text summarization, question answering, code generation, and scientific discovery across diverse domains [37]. Central to harnessing the full potential of these models is prompt engineering—the art and science of crafting effective prompts that guide LLMs to produce accurate, context-aware, and relevant outputs [2]. As LLMs become increasingly embedded in sectors like business, healthcare, and education, the demand for robust, adaptable, and efficient prompt engineering strategies has grown [12,15].

Recent advances have introduced a variety of prompt engineering paradigms, including zero-shot, few-shot, and chain-of-thought prompting, as well as automated frameworks that optimize prompts for specific tasks and domains [2]. These innovations address persistent challenges such as phrasing sensitivity, inconsistent responses, and the need for domain-specific knowledge integration [51]. Moreover, prompt engineering has shown promise in enhancing model reliability, reducing bias, and improving performance in knowledge-intensive and specialized fields [6,16].

This paper explores the evolving landscape of context-aware prompt engineering, highlighting key techniques, domain-specific adaptations, and emerging frameworks that empower users to unlock the transformative capabilities of LLMs across a wide array of real-world applications [12].

Prompt engineering has emerged as a critical skill for optimizing the performance of large language models (LLMs) across various domains, including healthcare and software development [1,11]. It involves crafting instructions to guide LLMs in generating desired outputs, with techniques categorized into instruction-based, information-based, reformulation, and metaphorical prompts [2]. Effective prompts can enhance LLM performance in tasks such as machine translation, question answering, and text generation [2]. A catalog of prompt patterns has been developed to address common problems when interacting with LLMs, providing reusable solutions analogous to software patterns [50]. Factors influencing prompt effectiveness include length, complexity, specificity, and context [2]. As LLMs become more accessible and widely used, particularly in healthcare, developing prompt engineering skills is increasingly important for professionals to improve their interactions with these AI tools [11].

Prompt engineering is a burgeoning field that plays a crucial role in enhancing the performance of large language models (LLMs) by crafting task-specific instructions, known as prompts, to guide model behavior without altering core parameters [2]. The quality of these prompts significantly impacts downstream task performance, as they provide the necessary context and instructions that enable LLMs to generate accurate and relevant outputs [2,3]. Crafting effective prompts across diverse domains presents challenges, including the need for complex reasoning to identify and rectify errors in prompts, and the difficulty in conveying clear, complete requirements for varied tasks [4,3]. Understanding user intent is paramount, as it ensures that the prompts align with the desired outcomes, thereby enhancing the efficacy of LLM outputs [3]. Motivated by these challenges, Smart Prompt aims to leverage LLMs for context-aware prompt engineering, focusing on dynamic prompt enhancement through intent understanding [5].




## 2. Related Work

### 2.1 Prompt Engineering Tools and Techniques

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

**Pattern Catalogs and Frameworks:** Catalogs of prompt patterns provide reusable solutions for structuring prompts, enabling users to address common challenges and adapt LLMs to various domains. These patterns can be combined for more complex tasks and are analogous to software design patterns [50].

**Prompt Engineering in Practice:** Prompt engineering is widely used to guide LLMs in generating desired outputs without altering model parameters [48,2]. Techniques include in-context learning, chain-of-thought prompting, and self-refinement, which offer flexibility and resource efficiency, especially in domains with limited computational resources [48,2].

**Domain-Knowledge Embedded Prompts:** Integrating domain-specific knowledge into prompts significantly improves LLM performance in specialized fields, such as chemistry and medicine, by enhancing accuracy and reducing hallucinations [16,18].

### 2.2 Instruction-Tuning and Fine-Tuning in LLMs

**Instruction-Tuning Frameworks:** Tools like FLAN, Alpaca, and LLaMA use instruction-tuning to adapt LLMs for specific tasks. For example, ToolLLM introduces a framework for tool-use by constructing large instruction-tuning datasets and fine-tuning LLaMA, enabling robust API interaction and generalization [47].

**Sample Design Engineering:** The design of input/output samples for fine-tuning (Sample Design Engineering) is critical for downstream performance, with systematic strategies outperforming heuristic approaches in complex tasks [49].

**Comparative Effectiveness:** Prompt engineering and fine-tuning each have strengths; prompt engineering offers flexibility, while fine-tuning can yield superior results for certain tasks, such as code generation, depending on the context and task complexity [52].

### 2.3 Domain-Specific LLM Applications

**Scientific and Medical Domains:** Domain-specific prompt engineering and instruction-tuning are essential for applications in chemistry, medicine, and wireless networks, where specialized terminology and requirements demand tailored approaches [16,48,18].

**Software Engineering:** LLMs are applied to code generation, summarization, and translation, with both prompt engineering and fine-tuning being actively compared for effectiveness [52].

### 2.4 Gaps and Limitations in Current Methods

**Personalization and Intent Parsing:** Current methods often lack deep personalization and dynamic intent parsing, limiting their adaptability to individual user needs and nuanced instructions [2,51].

**Sensitivity to Instruction Phrasing:** Instruction-tuned LLMs, especially in clinical domains, can be highly sensitive to prompt phrasing, affecting both performance and fairness [51].

**Reporting and Baselines:** Many studies do not report non-prompt baselines or key prompt engineering details, hindering reproducibility and systematic progress [18].

## 3. Fine-Tuning Methodology

Fine-tuning large language models like Mistral 7B using LoRA-based methods on accessible hardware (such as T4 GPUs) is a practical approach for adapting models to specialized tasks [42,44]. The following overview synthesizes recent research on fine-tuning methodology, training setup, and evaluation strategies for Mistral 7B, with a focus on frameworks and best practices.

### 3.1 Fine-Tuning Methodology

**Parameter-Efficient Fine-Tuning (PEFT):** LoRA (Low-Rank Adaptation) and QLoRA are widely used for efficient fine-tuning of Mistral 7B, enabling adaptation with reduced computational resources and memory requirements [42,44,46].

**Frameworks:** Commonly used frameworks include Hugging Face Transformers, TRL (Transformer Reinforcement Learning), PEFT, and Accelerate, which streamline the fine-tuning process and support integration with various hardware setups [44,46].

**Data Preparation:** Fine-tuning datasets are often curated for the target domain, such as legal, financial, or scientific texts. Data is typically formatted in JSONL, with tasks ranging from question-answer pairs to

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

sequence classification and translation [38,39,40,42,44,45].

### 3.2 Training Setup

| Parameter | Typical Values/Practices |
|---|---|
| Batch Size | Adjusted to GPU memory (e.g., 8–32) |
| Epochs | Ranges from a few (3–5) to several dozen, depending on dataset size and convergence (Jindal et al., 2024; Christodoulou, 2024) |
| Optimizer | Adam or AdamW optimizers are commonly used |
| Hardware | Single or multi-GPU setups (T4, RTX 4090, A100) (Jindal et al., 2024; Christodoulou, 2024) |

Training duration and resource allocation are tailored to the dataset and task complexity, with some models fine-tuned within 16–24 hours on a single high-end GPU [41,44].

### 3.3 Evaluation Strategy

**Automatic Metrics:** BLEU, BERTScore, and Macro-F1 are frequently used for quantitative evaluation, especially in translation, classification, and question-answering tasks [40,42,43,44,46].

**Human Evaluation:** In some studies, human raters assess prompt effectiveness, syntactic correctness, and domain relevance, particularly for specialized or open-ended tasks [38,42].

**Prompt Effectiveness:** Fine-tuned models are compared to base models and other LLMs (e.g., GPT-4, NLLB 3.3B) to assess improvements in accuracy, robustness, and domain adaptation [38,40,42,46].

### 3.4 Key Findings and Best Practices

**Instruction Fine-Tuning:** Instruction-tuned models maintain more robust performance and generalize better to unseen tasks compared to base model fine-tuning [39,45,46].

**Model Merging:** Combining single-task fine-tuned models with base models can further enhance zero-shot and domain-specific performance [39].

**Prompt Sensitivity:** Fine-tuned models are less sensitive to prompt variations, improving reliability in real-world applications [46].

## 4. Use Cases & Applications

Large language models (LLMs) are increasingly being applied across a variety of domains, offering advanced capabilities in code generation, image creation, academic writing, legal analysis, and healthcare support [28,30,31,33,34]. Research highlights both the strengths and current limitations of LLMs in these areas, as well as emerging frameworks and strategies to enhance their effectiveness.

### 4.1 Code Generation

**Framework and Component Suggestions:** LLMs can generate code from natural language descriptions, suggest frameworks, and recommend components for building applications. They are particularly effective when they can clarify ambiguous or incomplete requirements by asking follow-up questions, which improves code accuracy and quality [28,32,37].

**Collaborative and Self-Collaborative Approaches:** Multi-agent frameworks, where LLMs take on roles such as analyst, coder, and tester, significantly improve performance on complex tasks compared to single-agent approaches, boosting code correctness and handling repository-level challenges [29].

**Specialized Applications:** LLMs are used for data wrangling, safety-critical software, and even generating control logic from industrial diagrams, demonstrating versatility across software engineering and automation tasks [32,33,36].

**Code Understanding:** LLMs integrated into development environments help users understand code, explain APIs, and provide usage examples, aiding both students and professionals [35].

### 4.2 Academic Writing

**Enhancing Research Statements:** LLMs assist in generating and refining research problem statements, developing scripts for data analysis, and supporting

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

scientific writing. Their effectiveness varies across tools, with some issues in output integrity, but they show promise in increasing productivity for researchers [34].

### 4.3 Legal Applications

**Contract Formatting and Terminology:** LLMs are used for legal document analysis, contract review, case prediction, and summarization. Advanced frameworks like CoLE use intent identification and collaborative prompting to improve accuracy and relevance in legal advice, even handling colloquial queries and complex legal language [30,31].

**Trends and Challenges:** There is a growing interest in LLMs for legal tasks, with improvements in performance and methodological sophistication, though challenges remain in handling domain-specific terminology and logic [30].

### 4.4 Healthcare

Symptom Explanation and Diagnosis: While not directly covered in the provided abstracts, the methodologies used in legal and academic domains—such as intent extraction and prompt enhancement—are applicable to healthcare, where LLMs can structure and clarify symptom explanations or diagnostic prompts.

Enhanced prompts—those refined for clarity, specificity, and domain alignment—consistently outperform raw user prompts in both automated and human evaluations. Research across education, programming, and natural language tasks demonstrates that prompt enhancement leads to more helpful, clear, and contextually appropriate outputs.

### 4.5 Enhanced vs. Raw Prompt Performance

Enhanced prompts—those refined for clarity, specificity, and domain alignment—consistently outperform raw user prompts in both automated and human evaluations [20,21,22]. Research across education, programming, and natural language tasks demonstrates that prompt enhancement leads to more helpful, clear, and contextually appropriate outputs.

**Performance Gains:** Enhanced prompts, such as structured or matrix-form prompts, significantly improve task performance, comprehension, and integration of information compared to raw or less-structured prompts [20,24,25].

**Domain Adaptation:** Models using enhanced prompts are better at aligning outputs with domain-specific requirements, such as essay scoring or medical text classification, leading to higher accuracy and consistency [22,23,27].

| Domain | Raw Prompt Output | Enhanced Prompt Output |
|---|---|---|
| Education | Generic, less detailed questions | Context-rich, reasoning-based questions |
| Essay Scoring | Inconsistent trait assessment | Higher consistency, better trait alignment |
| Medical | Lower classification accuracy | Improved accuracy, better label relevance |

**Human Rating:** Human evaluators rate outputs from enhanced prompts as more helpful and clearer than those from raw prompts. For example, in educational question generation, longer, more descriptive prompts yield questions that are more coherent and contextually relevant [21,24,26].

## 5. Context-Aware Generation Systems

Context-aware generation systems are increasingly used in real-world applications for their ability to adapt outputs based on user intent, environment, and domain [53,54,55]. These systems offer notable strengths in generalization and context sensitivity, but also face challenges related to data dependency, bias, and ethical risks.

### 5.1 Strengths: Domain Generalization & Context-Aware Generation

**Domain Generalization:** Context-aware frameworks can incorporate multiple data sources and adapt to a wide range of domains, improving recommendation accuracy, personalization, and decision support across diverse applications such as IoT, smart environments, and content delivery [53,54,55,57].

**Context-Aware Generation:** By dynamically integrating relevant contextual information, these systems generate outputs that are more accurate, relevant, and personalized. Multi-channel retrieval and context modeling enable real-time adaptation to evolving user needs and environments, enhancing

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

usability in complex, dynamic settings [54,55,56,57,58].

### 5.2 Limitations: Data Dependency & Bias

**Dependency on Training Data Variety:** The effectiveness of context-aware systems is highly dependent on the diversity and quality of training data. Limited or biased data can restrict generalization and lead to suboptimal or skewed outputs, especially in underrepresented domains or scenarios [53,56,58].

**Bias in Suggestions:** Context-aware models may inadvertently reinforce existing biases present in their training data, affecting fairness and reliability in sensitive applications such as healthcare, legal, or public services [56,60,61].

**Integration Challenges:** Many systems struggle to integrate data from outside their primary domain, limiting cross-domain contextualization and reducing the potential for richer, more holistic outputs [57].

### 5.3 Real-World Usability: API and UI Integration

**API Use in Prompt Tools:** Context-aware systems are increasingly deployed via APIs, enabling integration into prompt engineering tools, chat interfaces, and collaborative environments for real-time, user-driven interactions [54,57,59].

**Smart Environments:** These systems support a variety of applications, from smart cities and IoT to robotic services and vehicle trajectory prediction, demonstrating scalability and adaptability in real-world deployments [55,57,58,59].

### 5.4 Ethical Considerations: Hallucination & Disclaimers

**Hallucination Risk:** Context-aware models can generate plausible but incorrect or misleading outputs, especially when context is ambiguous or data is insufficient, raising concerns in high-stakes domains [56,60].

**Legal/Medical Disclaimers:** The use of context-aware generation in legal or medical settings necessitates clear disclaimers and safeguards to prevent misuse and ensure users are aware of the system's limitations and potential for error [60,61].

## 6. Summary of Findings

Smart Prompt advances LLM prompt engineering by introducing a modular architecture that systematically parses user input, extracts intent, classifies domain, and dynamically enhances prompts for improved clarity, relevance, and domain alignment [19]. This approach leverages prompt engineering as a critical skill for maximizing LLM effectiveness across fields such as healthcare, education, entrepreneurship, and scientific research, enabling more accurate, context-aware, and innovative outputs [11,13,14,15,16,17,18,19].

Smart Prompt's design reflects the growing recognition that prompt engineering—when combined with frameworks for prompt management and postprocessing—can bridge the gap between raw user queries and high-quality, domain-specific LLM responses [12,16,17,19].

### 6.1 Smart Prompt's Innovation in LLM Prompt Engineering

**Systematic Prompt Enhancement:** By integrating intent extraction, domain classification, and prompt enrichment, Smart Prompt addresses the limitations of ad hoc, trial-and-error prompt development, moving toward a more structured, software engineering-inspired methodology [19].

**Domain-Specific Adaptation:** Embedding domain knowledge into prompts significantly boosts LLM performance in specialized areas, such as scientific discovery and medical applications, reducing hallucinations and improving accuracy [16,18].

**User-Centric Design:** Smart Prompt's architecture supports real-time, context-aware interactions, empowering users to co-create and refine prompts, which fosters more engaging and equitable experiences in both professional and educational settings [11,15,17].

**Extensibility and Innovation:** The modular design allows for easy adaptation to new domains and tasks, supporting innovation and the rapid deployment of LLM-powered solutions in diverse contexts [12,14,19].

## 7. Future Work

**Larger and More Diverse Datasets:** Expanding training and evaluation datasets will improve model robustness, generalizability, and the ability to handle nuanced, domain-specific queries [16,18].

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-6 | June - 2025 | Page 62-71

**UI-Based Real-Time Prompt Tweaking:** Developing user interfaces for interactive, real-time prompt refinement will further empower users and enhance the adaptability of LLM systems [15,17].

**Cross-Model Comparison:** Systematic benchmarking across different LLMs and prompt engineering strategies will help identify best practices, address current reporting gaps, and drive progress in promptware engineering [16,18,19].

## 8. Conclusion

Smart Prompt exemplifies the next generation of LLM prompt engineering by combining structured, modular design with domain-specific adaptation and user-centric interaction [19]. Context-aware generation excels at adapting to diverse domains and providing personalized, contextually relevant outputs, making it valuable for real-world applications [54,55]. However, its effectiveness is limited by data quality, integration challenges, and ethical risks such as bias and hallucination [56,60]. Careful design, diverse training data, and transparent user communication are essential for safe and effective deployment.

Continued research into larger datasets, interactive interfaces, and systematic evaluation will be essential for realizing the full potential of prompt engineering and advancing the capabilities of LLM-powered applications [16,18,19].

## References

[1] D. Lamba, "The Role of Prompt Engineering in Improving Language Understanding and Generation," International Journal For Multidisciplinary Research, vol. 6, no. 6, 2024, doi: 10.36948/ijfmr.2024.v06i06.32232.

[2] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," arXiv preprint arXiv:2402.07927, 2024, doi: 10.48550/arxiv.2402.07927.

[3] Q. Ma, W. Peng, H. Shen, K. R. Koedinger, and T. Wu, "What You Say = What You Want? Teaching Humans to Articulate Requirements for LLMs," arXiv preprint arXiv:2409.08775, 2024, doi: 10.48550/arxiv.2409.08775.

[4] Q. Ye, M. Axmed, R. Pryzant, and F. Khani, "Prompt Engineering a Prompt Engineer," arXiv preprint arXiv:2311.05661, 2023, doi: 10.48550/arxiv.2311.05661.

[5] N. Wu, "Expanding Horizons in Prompt Engineering: Techniques, Frameworks, and Challenges," Indian Scientific Journal Of Research In Engineering And Management, vol. 9, no. 1, pp. 1-6, 2025, doi: 10.55041/ijsrem40635.

[6] S. Dwivedi, S. Dwivedi, and S. Ghosh, "Breaking the Bias: Gender Fairness in LLMs Using Prompt Engineering and In-Context Learning," Rupkatha Journal on Interdisciplinary Studies in Humanities, vol. 15, no. 4, 2023, doi: 10.21659/rupkatha.v15n4.10.

[7] M. Hewing and V. Leinhos, "The Prompt Canvas: A Literature-Based Practitioner Guide for Creating Effective Prompts in Large Language Models," arXiv preprint arXiv:2412.05127, 2024, doi: 10.48550/arxiv.2412.05127.

[8] M. F. Russe, M. Reisert, F. Bamberg, and A. Rau, "Improving the use of LLMs in radiology through prompt engineering: from precision prompts to zero-shot learning," RoFo : Fortschritte Auf Dem Gebiete Der Rontgenstrahlen Und Der Nuklearmedizin, vol. 196, no. 11, pp. 1166-1170, 2024, doi: 10.1055/a-2264-5631.

[9] D. Park, G.-T. An, C. Kamyod, and C. G. Kim, "A Study on Performance Improvement of Prompt Engineering for Generative AI with a Large Language Model," Journal of Web Engineering, pp. 1187-1206, 2024, doi: 10.13052/jwe1540-9589.2285.

[10] W. Jin, B. Wu, T. Tao, X. Wang, B. Zhao, Y. Gao, and N. Wang, "Veracity-Oriented Context-Aware Large Language Models–Based Prompting Optimization for Fake News Detection," International Journal of Intelligent Systems, vol. 2025, no. 1, 2025, doi: 10.1155/int/5920142.

[11] B. Meskó, "Prompt Engineering as an Important Emerging Skill for Medical Professionals: Tutorial," Journal of Medical Internet Research, vol. 25, 2023, doi: 10.2196/50638.

[12] B. Xiao, B. Kantarci, J. Kang, D. Niyato, and M. Guizani, "Efficient Prompting for LLM-Based

Generative Internet of Things," IEEE Internet of Things Journal, vol. 12, pp. 778-791, 2024, doi: 10.1109/JIOT.2024.3470210.

[13] M. Thanasi-Boçe and J. Hoxha, "From ideas to ventures: building entrepreneurship knowledge with LLM, prompt engineering, and conversational agents," Education and Information Technologies, vol. 29, pp. 24309-24365, 2024, doi: 10.1007/s10639-024-12775-z.

[14] L. Sundberg and J. Holmström, "Innovating by prompting: How to facilitate innovation in the age of generative AI," Business Horizons, 2024, doi: 10.1016/j.bushor.2024.04.014.

[15] W. Cain, "Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education," TechTrends, 2023, doi: 10.1007/s11528-023-00896-0.

[16] H. Liu, H. Yin, Z. Luo, and X. Wang, "Integrating chemistry knowledge in large language models via prompt engineering," Synthetic and Systems Biotechnology, vol. 10, pp. 23-38, 2024, doi: 10.1016/j.synbio.2024.07.004.

[17] L. Sreedhar, "Speak the Language of AI: Mastering Prompt Engineering for LLMs," in Proceedings of the International Conference on Industrial Engineering and Operations Management, 2024, doi: 10.46254/in04.20240201.

[18] J. Zaghir, M. Naguib, M. Bjelogrlic, A. Névéol, X. Tannier, and C. Lovis, "Prompt Engineering Paradigms for Medical Applications: Scoping Review," Journal of Medical Internet Research, vol. 26, 2024, doi: 10.2196/60501.

[19] Z. Chen, C. Wang, W. Sun, G. Yang, X. Liu, J. Zhang, and Y. Liu, "Promptware Engineering: Software Engineering for LLM Prompt Development," 2025.

[20] L. Guo, "The Effects of the Format and Frequency of Prompts on Source Evaluation and Multiple-Text Comprehension," Reading Psychology, vol. 44, pp. 358-387, 2022, doi: 10.1080/02702711.2022.2156949.

[21] S. Maity, A. Deroy, and S. Sarkar, "Harnessing the Power of Prompt-based Techniques for Generating School-Level Questions using Large Language Models," in Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, 2023, doi: 10.1145/3632754.3632755.

[22] J. Xu, J. Liu, M. Lin, J. Lin, S. Yu, L. Zhao, and J. Shen, "EPCTS: Enhanced Prompt-Aware Cross-Prompt Essay Trait Scoring," Neurocomputing, vol. 621, p. 129283, 2025, doi: 10.1016/j.neucom.2024.129283.

[23] Y. Wang, L. Zhou, W. Zhang, F. Zhang, and Y. Wang, "A soft prompt learning method for medical text classification with simulated human cognitive capabilities," Artificial Intelligence Review, vol. 58, p. 118, 2025, doi: 10.1007/s10462-025-11121-0.

[24] A. Garg, N. Soodhani, and R. Rajendran, "Enhancing data analysis and programming skills through structured prompt training: The impact of generative AI in engineering education," Computers and Education: Artificial Intelligence, 2025, doi: 10.1016/j.caeai.2025.100380.

[25] S. Lee, A. Bahukhandi, D. Liu, and K. *, "Towards Dataset-Scale and Feature-Oriented Evaluation of Text Summarization in Large Language Model Prompts," IEEE Transactions on Visualization and Computer Graphics, vol. 31, pp. 481-491, 2024, doi: 10.1109/TVCG.2024.3456398.

[26] N. Wu, "A Comparative Analysis of Human-Written vs. AI-Generated Prompts for Task Executions," International Journal of Scientific Research in Engineering and Management, 2024, doi: 10.55041/ijsrem39321.

[27] Y. Wang and Y. Zhao, "Metacognitive Prompting Improves Understanding in Large Language Models," arXiv preprint arXiv:2308.05342, 2023, doi: 10.48550/arXiv.2308.05342.

[28] J. Wu and F. Fard, "HumanEvalComm: Benchmarking the Communication Competence of Code Generation for LLMs and LLM Agent," ACM Transactions on Software Engineering and Methodology, 2024, doi: 10.1145/3715109.

[29] Y. Dong, X. Jiang, Z. Jin, and G. Li, "Self-collaboration Code Generation via ChatGPT," ACM Transactions on Software Engineering and Methodology, 2023, doi: 10.1145/3672459.

[30] M. Siino, M. Falco, D. Croce, and P. Rosso, "Exploring LLMs Applications in Law: A Literature

Review on Current Legal NLP Approaches," IEEE Access, vol. 13, pp. 18253-18276, 2025, doi: 10.1109/ACCESS.2025.3533217.

[31] B. Li, S. Fan, S. Zhu, and L. Wen, "CoLE: A collaborative legal expert prompting framework for large language models in law," Knowledge-Based Systems, vol. 311, p. 113052, 2025, doi: 10.1016/j.knosys.2025.113052.

[32] X. Li and T. Döhmen, "Towards Efficient Data Wrangling with LLMs using Code Generation," in Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning, 2024, doi: 10.1145/3650203.3663334.

[33] M. Liu, J. Wang, T. Lin, Q. *, Z. Fang, and Y. Wu, "An Empirical Study of the Code Generation of Safety-Critical Software Using LLMs," Applied Sciences, 2024, doi: 10.3390/app14031046.

[34] M. Nejjar, L. Zacharias, F. Stiehle, and I. Weber, "LLMs for Science: Usage for Code Generation and Data Analysis," arXiv preprint arXiv:2311.16733, 2023, doi: 10.48550/arXiv.2311.16733.

[35] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers, "Using an LLM to Help with Code Understanding," in 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE), pp. 1184-1196, 2023, doi: 10.1145/3597503.3639187.

[36] H. Koziolek and A. Koziolek, "LLM-based Control Code Generation using Image Recognition," in 2024 IEEE/ACM International Workshop on Large Language Models for Code (LLM4Code), pp. 38-45, 2023, doi: 10.48550/arXiv.2311.10401.

[37] J. Jiang, F. Wang, J. Shen, S. Kim, and S. Kim, "A Survey on Large Language Models for Code Generation," arXiv preprint arXiv:2406.00515, 2024, doi: 10.48550/arXiv.2406.00515.

[38] D. Doumanas, A. Soularidis, D. Spiliotopoulos, C. Vassilakis, and K. Kotis, "Fine-Tuning Large Language Models for Ontology Engineering: A Comparative Analysis of GPT-4 and Mistral," Applied Sciences, 2025, doi: 10.3390/app15042146.

[39] S. Fatemi, Y. Hu, and M. Mousavi, "A Comparative Analysis of Instruction Fine-Tuning Large Language Models for Financial Text Classification," ACM Transactions on Management Information Systems, 2025, doi: 10.1145/3706119.

[40] M. Anam, A. Akbar, and H. Saputro, "QnA Chatbot with Mistral 7B and RAG method: Traffic Law Case Study," Lontar Komputer : Jurnal Ilmiah Teknologi Informasi, 2025, doi: 10.24843/lkjiti.2024.v15.i03.p06.

[41] A. Jindal, P. Rajpoot, and A. Parikh, "Birbal: An efficient 7B instruct-model fine-tuned with curated datasets," *arXiv preprint arXiv:2403.02247*, 2024, doi: 10.48550/arXiv.2403.02247.

[42] Y. Moslem, R. Haque, and A. Way, "Fine-tuning Large Language Models for Adaptive Machine Translation," *arXiv preprint arXiv:2312.12740*, 2023, doi: 10.48550/arXiv.2312.12740.

[43] R. Bayraktar, B. Sarıtürk, and M. Erdem, "Assessing Fine-Tuning Efficacy in LLMs: A Case Study with Learning Guidance Chatbots," *International Journal of Innovative Science and Research Technology (IJISRT)*, 2024, doi: 10.38124/ijisrt/ijisrt24may1600.

[44] C. Christodoulou, "NLPDame at ClimateActivism 2024: Mistral Sequence Classification with PEFT for Hate Speech, Targets and Stance Event Detection," pp. 96-104, 2024.

[45] S. Ahuja, K. Tanmay, H. Chauhan, B. Patra, K. Aggarwal, T. Dhamecha, M. Choudhary, V. Chaudhary, and S. Sitaram, "sPhinX: Sample Efficient Multilingual Instruction Fine-Tuning Through N-shot Guided Prompting," *arXiv preprint arXiv:2407.09879*, 2024, doi: 10.48550/arXiv.2407.09879.

[46] M. Topal, A. Bozanta, and A. Basar, "Sentiment Analysis with LLMs: Evaluating QLoRA Fine-Tuning, Instruction Strategies, and Prompt Sensitivity," in *2024 34th International Conference on Collaborative Advances in Software and Computing (CASCON)*, pp. 1-10, 2024, doi: 10.1109/CASCON62161.2024.10838185.

[47] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, and M. Sun, "ToolLLM: Facilitating Large Language Models to

Master 16000+ Real-world APIs," *arXiv preprint arXiv:2307.16789*, 2023, doi: 10.48550/arXiv.2307.16789.

[48] H. Zhou, C. Hu, D. Yuan, Y. Yuan, D. Wu, X. Chen, H. Tabassum, and X. Liu, "Large Language Models (LLMs) for Wireless Networks: An Overview from the Prompt Engineering Perspective," *arXiv preprint arXiv:2411.04136*, 2024, doi: 10.48550/arXiv.2411.04136.

[49] B. Guo, H. Wang, W. Xiao, H. Chen, Z. Lee, S. Han, and H. Huang, "Sample Design Engineering: An Empirical Study of What Makes Good Downstream Fine-Tuning Samples for LLMs," *arXiv preprint arXiv:2404.13033*, 2024, doi: 10.48550/arXiv.2404.13033.

[50] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. Schmidt, "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," *arXiv preprint arXiv:2302.11382*, 2023.

[51] A. Arroyo, M. Munnangi, J. Sun, K. Zhang, D. McInerney, B. Wallace, and S. Amir, "Open (Clinical) LLMs are Sensitive to Instruction Phrasings," pp. 50-71, 2024, doi: 10.48550/arXiv.2407.09429.

[52] J. Shin, C. Tang, T. Mohati, M. Nayebi, S. Wang, and H. Hemmati, "Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated Software Engineering Tasks," *arXiv preprint arXiv:2310.10508*, 2023, doi: 10.48550/arXiv.2310.10508.

[53] B. Hidasi and D. Tikk, "General factorization framework for context-aware recommendations," *Data Mining and Knowledge Discovery*, vol. 30, pp. 342-371, 2014, doi: 10.1007/s10618-015-0417-y.

[54] P. Buddiga, "Scaling Intelligent Systems with Multi-Channel Retrieval Augmented Generation (RAG): A robust Framework for Context Aware Knowledge Retrieval and Text Generation," in *2024 Global Conference on Communications and Information Technologies (GCCIT)*, pp. 1-10, 2024, doi: 10.1109/GCCIT63234.2024.10862867.

[55] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, pp. 414-454, 2013, doi: 10.1109/SURV.2013.042313.00197.

[56] I. Sarker, A. Colman, J. Han, and P. Watters, "Context-Aware Machine Learning System: Applications and Challenging Issues," in *Context-Aware Machine Learning and Mobile Data Analytics*, 2021, doi: 10.1007/978-3-030-88530-4_10.

[57] A. Bazan-Muñoz, G. Ortiz, J. Augusto, and A. De Prado, "Taxonomy and software architecture for real-time context-aware collaborative smart environments," *Internet of Things*, vol. 26, p. 101160, 2024, doi: 10.1016/j.iot.2024.101160.

[58] D. Chang and B. Han, "Knowledge-based Visual Context-Aware Framework for Applications in Robotic Services," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 1-9, 2023, doi: 10.1109/WACVW58289.2023.00012.

[59] P. Xu, J. Hayet, and I. Karamouzas, "Context-Aware Timewise VAEs for Real-Time Vehicle Trajectory Prediction," *IEEE Robotics and Automation Letters*, vol. 8, pp. 5440-5447, 2023, doi: 10.1109/LRA.2023.3295990.

[60] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, pp. 161-180, 2010, doi: 10.1016/j.pmcj.2009.06.002.

[61] C. Xu, "Context aware mobility in Internet of Things enabling technologies, applications, and challenges," *Transactions on Emerging Telecommunications Technologies*, vol. 33, 2022, doi: 10.1002/ett.4624.