how to writte AI prompts to get best results?

To get the best results from AI, use specific, detailed prompts that include explicit instructions, contextual information, and systematic task decomposition.

Abstract

Effective prompts for artificial intelligence require clarity, specificity, and context. Several studies show that open-ended, detailed prompts—with explicit instructions and stepwise task decomposition—yield higher accuracy and reduced ambiguity over vague or restrictive directions. Retrieval methods that select semantically similar or complex in-context examples have produced performance gains of roughly 44–45%. Automated optimization techniques—such as gradient-based refinement, reinforcement learning–driven rewriting, and Monte Carlo tree search—improve outcomes by 6–80%, particularly in natural language tasks. In domains like code and image generation, incorporating chain-of-thought processes and precise prompt anatomy leads to measurable improvements, with reported enhancements in reasoning and output quality (e.g., 3–74% increases in CodeBLEU scores). These results support a design approach that combines clear, detailed instructions with context-aware and automated refinement methods to obtain the best AI prompt performance.

Paper search

Using your research question "how to writte AI prompts to get best results?", we searched across over 126 million academic papers from the Semantic Scholar corpus. We retrieved the 50 papers most relevant to the query.

Screening

We screened in papers that met these criteria:

- **Prompt Engineering Focus**: Does the study examine prompt engineering techniques for AI language models?
- **LLM Relevance**: Does the research involve large language models (LLMs)?
- **Measurable Outcomes**: Does the study include measurable outcomes that demonstrate prompt effectiveness?
- **Evidence Type**: Does the study present empirical data or documented case study evidence?
- **Comparative Analysis**: Does the study compare different prompting strategies or evaluate specific prompt engineering techniques?
- **Scientific Rigor**: Does the study present original research with empirical evidence (rather than just opinions or editorials)?
- **Research Focus**: Does the study include prompt engineering components (rather than focusing solely on AI model architecture or training methods)?

We considered all screening questions together and made a holistic judgement about whether to screen in each paper.

Data extraction

We asked a large language model to extract each data column below from each paper. We gave the model the extraction instructions shown below for each column.

- **Study Design Type**:

  Identify the primary type of study design used:

- Experimental study
- Comparative study
- Theoretical/conceptual study
- Empirical analysis

Look in the methods or introduction section. If multiple design elements are present, list the most prominent type. If unclear, note "design type not clearly specified".

- **Prompt Engineering Approach**:

  Describe the specific approach used for prompt engineering:

- Retrieval-based selection
- Manual crafting
- Gradient-based optimization
- Guided prompt creation

Extract the precise methodology from the methods or results section. If multiple approaches are used, list all relevant approaches. Include specific techniques or strategies mentioned for improving prompt quality.

- **Prompt Characteristics Analyzed**:

  List the specific characteristics or dimensions of prompts that were examined:

- Clarity
- Detail level
- Semantic similarity
- Relevance
- Aesthetic appeal
- Creativity

Extract from methods, results, or discussion sections. Include any quantitative or qualitative metrics used to assess these characteristics. If specific criteria were used to evaluate prompts, note those explicitly.

- **Comparative Conditions**:

  Identify the comparison groups or conditions in the study:

- Random prompt selection vs. guided selection
- Supervised vs. unsupervised prompt creation
- Different prompt engineering techniques

Extract from the methods section. Clearly describe the different conditions being compared, including any control or baseline conditions. If no explicit comparison was made, note "No comparative conditions".

- **Key Performance Outcomes**:

  Document the primary outcomes or performance metrics:

- Improvement percentages
- Task-specific performance gains

- Qualitative assessment results

Extract from results and discussion sections. Include specific numerical improvements, statistical significance, and context of the outcomes. If multiple outcomes were measured, list all significant findings.

Results

Characteristics of Included Studies

Study	Study Focus	Application Domain	Methodology	Key Innovation	Full text retrieved
Arora et al., 2022	Aggregation of multiple imperfect prompts (AMA)	General Natural Language Processing (NLP) tasks (Question Answering (QA), Natural Language Understanding (NLU))	Comparative study across large language models and benchmarks	Recursive large language model-guided prompt transformation and weak supervision aggregation	Yes
Gonen et al., 2022	Perplexity-based prompt selection (SPELL)	General Natural Language Processing tasks	Empirical analysis	Automated prompt expansion and selection via lowest perplexity	Yes
Pryzant et al., 2023	Automatic Prompt Optimization (APO, ProTeGi)	Natural Language Processing benchmarks, jailbreak detection	Experimental study	Gradient-based prompt optimization with beam search and bandit selection	Yes
Wang et al., 2023	PromptAgent for expert-level prompt optimization	General Natural Language Processing, domain-specific, BIG-Bench Hard	Empirical analysis	Strategic planning via Monte Carlo tree search and error feedback	Yes

Study	Study Focus	Application Domain	Methodology	Key Innovation	Full text retrieved
Liu et al., 2021	Retrieval-based in-context example selection	Natural Language Understanding, Natural Language Generation, Question Answering, table-to-text	Empirical analysis	Semantic similarity and complexity-based example retrieval	Yes
Ahmad and Ruslan, 2024	Guided vs. unguided prompt creation for image generation	Text-to-image (creative industries, education)	Comparative study	Prompt anatomy, expert panel evaluation of prompt quality	No
Liu et al., 2023	Prompt design for code generation with ChatGPT	Code generation (Text-to-Code, Code-to-Code)	Experimental study	Manual crafting, chain-of-thought, multi-step optimization	Yes
Kong et al., 2024	PRewrite: Reinforcement Learning-based prompt rewriting	Diverse Natural Language Processing benchmarks	Experimental study	Automated prompt rewriting with Reinforcement Learning (Proximal Policy Optimization), infer-ence/search strategies	Yes
Mishra et al., 2021	Reframing instructional prompts, guidance levels	Conceptual math Question Answering, Retrieval-Augmented Generation	Empirical analysis	Manual reframing, guidance conditions, retrieval-augmented generation	Yes

Wen et al., 2023	PEZ: Gradient-based hard prompt optimization	Text-to-image, text-to-text	Experimental study	Gradient-based discrete optimization for interpretable hard prompts	Yes

Application Domain:

- Five studies focused on general Natural Language Processing tasks.
- Three studies used Natural Language Processing benchmarks (including BIG-Bench Hard).
- Two studies addressed text-to-image tasks.
- One study focused on code generation.
- One study each addressed text-to-text, Natural Language Understanding, Natural Language Generation, table-to-text, jailbreak detection, domain-specific tasks, creative industries, education, Retrieval-Augmented Generation, and math Question Answering.
- Two studies focused on Question Answering tasks.

Methodology:

- Four studies used empirical analysis (focused on observed data).
- Four studies used experimental study designs (involving controlled interventions).
- Two studies used comparative study designs (directly comparing different methods).

Key Innovation:

- Automated prompt selection or optimization methods were reported in seven studies, including large language model-guided, perplexity-based, retrieval-based, reinforcement learning-based, and gradient-based approaches.
- Two studies used gradient-based optimization.
- One study used reinforcement learning-based optimization.
- One study used retrieval-based example selection.
- Two studies used manual prompt engineering (manual crafting, reframing, or guided/unguided prompt creation).
- One study used expert panel evaluation of prompt quality.
- One study used chain-of-thought or multi-step prompt design.
- One study used prompt anatomy analysis.
- One study used Monte Carlo tree search for strategic planning.
- One study used weak supervision aggregation.
- One study used retrieval-augmented generation.

We did not find mention of studies focusing exclusively on domains outside Natural Language Processing, code, or image generation, nor did we find mention of studies using methodologies other than empirical, experimental, or comparative designs.

Thematic Analysis

Fundamental Prompt Design Principles

- Structure and Formatting Guidelines:
  - Across studies, prompt clarity and specificity were repeatedly identified as critical for optimal artificial intelligence performance.
  - Open-ended, detailed, and contextually relevant prompts were reported to outperform restrictive or vague instructions.
- Clarity and Specificity Requirements:
  - Explicit instructions, stepwise decomposition, and reframing complex tasks into simpler sub-tasks were emphasized as important for effective prompt design.
- Context Integration Methods:
  - Retrieval-based selection of semantically similar or complex in-context examples was found to significantly enhance model performance, especially in few-shot and in-context learning settings.

Advanced Optimization Techniques

- Strategic Planning Approaches:
  - Methods such as PromptAgent (using Monte Carlo tree search) and PRewrite (using reinforcement learning-based rewriting) demonstrated that systematic exploration and iterative refinement of prompts can yield expert-level, domain-insightful instructions.
- Example Selection Methods:
  - Complexity-based and semantic similarity-based retrieval of in-context examples were reported as highly effective, with large performance gains in Natural Language Understanding and Natural Language Generation tasks.
- Chain-of-Thought Implementation:
  - Incorporating chain-of-thought reasoning, either manually or via guided prompt creation, was found to improve performance in code generation and complex reasoning tasks.

Domain-Specific Considerations

- Code Generation Optimization:
  - For code generation, prompt design benefited from explicit requirements, chain-of-thought strategies, and iterative refinement, leading to substantial improvements in CodeBLEU scores.
- Image Generation Requirements:
  - In text-to-image tasks, prompt anatomy (clarity, detail, relevance, aesthetic appeal, inventiveness) and guided creation were reported as essential for high-quality outputs.
- General Text Task Adaptations:
  - For general Natural Language Processing tasks, aggregation of multiple imperfect prompts, perplexity-based selection, and reframing for clarity and relevance were effective strategies.

Theme	Key Findings	Implementation Strategies	Success Factors
Clarity & Specificity	Open-ended, detailed prompts outperform restrictive ones	Manual crafting, reframing, explicit instructions	Improved accuracy, reduced ambiguity

Theme	Key Findings	Implementation Strategies	Success Factors
Context Integration	Retrieval-based, complexity-based example selection yields large gains	Semantic similarity retrieval, complexity metrics	Robustness, generalization
Optimization Techniques	Reinforcement learning, gradient-based, and strategic planning methods outperform manual approaches	Reinforcement learning (PRewrite), gradient descent (ProTeGi, PEZ), Monte Carlo tree search (PromptAgent)	Consistent performance gains, interpretable prompts
Chain-of-Thought	Stepwise reasoning enhances complex task performance	Chain-of-thought prompt design, multi-step optimization	Higher correctness, better reasoning
Domain Adaptation	Domain-specific detail and prompt anatomy critical in code/image tasks	Prompt anatomy, explicit requirements, style transfer	Task-specific quality, user satisfaction

Key Findings:

- Two studies reported that open-ended, detailed prompts outperform restrictive ones.
- Two studies reported that retrieval-based or complexity-based example selection yields large gains.
- Four studies reported that reinforcement learning, gradient-based, and strategic planning methods outperform manual approaches.
- Two studies reported that stepwise (chain-of-thought) reasoning enhances complex task performance.
- Three studies reported that domain-specific detail and prompt anatomy are critical in code/image tasks.

Implementation Strategies:

- Manual crafting, reframing, and explicit instructions were used in two studies.
- Semantic similarity retrieval and complexity metrics were used in two studies.
- Reinforcement learning (PRewrite) was used in at least one study; gradient descent (ProTeGi, PEZ) in at least two studies; and Monte Carlo tree search (PromptAgent) in at least one study.
- Chain-of-thought prompt design and multi-step optimization were used in two studies.
- Prompt anatomy, explicit requirements, and style transfer were used in three studies.

We did not find mention in the table about studies that did not support these findings, nor about the total number of studies reviewed.

Cross-Domain Best Practices

Universal Guidelines

- Common Success Factors:

  - Clarity, detail, and relevance were foundational across all domains.
  - Automated optimization (reinforcement learning, gradient-based, strategic planning) and retrieval-based example selection were consistently effective.
- Transferable Techniques:
  - Methods such as chain-of-thought prompting, semantic similarity retrieval, and iterative prompt refinement generalized well across text, code, and image generation tasks.
- Validation Methods:
  - Both quantitative (accuracy, Bilingual Evaluation Understudy (BLEU), Contrastive Language–Image Pretraining (CLIP) scores) and qualitative (expert panel, human preference) assessments were used, though standardization was lacking.

Practice	Application Areas	Effectiveness Indicators	Limitations
Guided/Automated Prompt Optimization (reinforcement learning, gradient, Monte Carlo tree search)	Text, code, image generation	6–80% improvement, expert-level prompts, interpretable outputs	Requires computational resources, may overfit to specific tasks
Retrieval-Based Example Selection	In-context learning, Natural Language Understanding/Natural Language Generation	44–45% gains, robust across models	Dependent on quality of retrieval module, may not generalize to all tasks
Chain-of-Thought Prompting	Code, complex reasoning	3–74% CodeBLEU improvement, higher correctness	May increase prompt length, not always preferred by users
Prompt Anatomy & Guided Creation	Image generation, creative tasks	Improved clarity, detail, aesthetic appeal	Qualitative assessment, subjective criteria
Aggregation of Multiple Prompts	General Natural Language Processing tasks	10.2% average lift, matches/exceeds larger models	May require ensemble methods, increased inference cost

Across the five studies summarized in this table, we found the following application areas for prompt engineering practices:

- Code generation: two studies
- Image generation: two studies
- Text generation: one study
- In-context learning: one study
- Natural Language Understanding/Natural Language Generation: one study
- Complex reasoning: one study
- Creative tasks: one study
- General Natural Language Processing tasks: one study

Effectiveness Indicators:

- Four of these five studies reported quantitative improvements, with gains ranging from 3% to 80% depending on the metric and task.

- All five studies reported at least one qualitative improvement, such as expert-level prompts, improved clarity or detail, higher correctness, interpretable outputs, robustness across models, or matching/exceeding larger models.

We did not find mention of studies in this table that lacked either a quantitative or qualitative effectiveness indicator.

References

Chao Liu, Xuanlin Bao, Hongyu Zhang, Neng Zhang, Haibo Hu, Xiaohong Zhang, and Meng Yan. "Improving ChatGPT Prompt for Code Generation." *arXiv.org*, 2023.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. "Demystifying Prompts in Language Models via Perplexity Estimation." *Conference on Empirical Methods in Natural Language Processing*, 2022.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, L. Carin, and Weizhu Chen. "What Makes Good In-Context Examples for GPT-3?" *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*, 2021.

Noor Wahyuni Ahmad, and Suzana Ruslan. "Crafting Effective Prompts: A Guideline for Successful Image Generation." *International Conference on System Engineering and Technology*, 2024.

Reid Pryzant, Dan Iter, Jerry Li, Y. Lee, Chenguang Zhu, and Michael Zeng. "Automatic Prompt Optimization with "Gradient Descent" and Beam Search." *Conference on Empirical Methods in Natural Language Processing*, 2023.

Simran Arora, A. Narayan, Mayee F. Chen, Laurel J. Orr, Neel Guha, Kush S. Bhatia, Ines Chami, Frederic Sala, and Christopher R'e. "Ask Me Anything: A Simple Strategy for Prompting Language Models." *International Conference on Learning Representations*, 2022.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. "Reframing Instructional Prompts to GPTk's Language." *Findings*, 2021.

Weize Kong, Spurthi Amba Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. "PRewrite: Prompt Rewriting with Reinforcement Learning." *Annual Meeting of the Association for Computational Linguistics*, 2024.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P. Xing, and Zhiting Hu. "PromptAgent: Strategic Planning with Language Models Enables Expert-Level Prompt Optimization." *International Conference on Learning Representations*, 2023.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and T. Goldstein. "Hard Prompts Made Easy: Gradient-Based Discrete Optimization for Prompt Tuning and Discovery." *Neural Information Processing Systems*, 2023.