# From Instruction to Output: The Role of Prompting in Modern NLG

**Anonymous ACL submission**


## Abstract

Prompt engineering has emerged as an integral technique for extending the strengths and abilities of Large Language Models (LLMs) to gain significant performance gains in various Natural Language Processing (NLP) tasks. This approach, which requires instructions to be composed in natural language to bring out the knowledge from LLMs in a structured way, has driven breakthroughs in various NLP tasks. Yet, there is still no structured framework or coherent understanding of the varied prompt-engineering methods and techniques, particularly in the field of Natural Language Generation (NLG).

This survey aims to help fill that gap by outlining recent developments in prompt engineering, and their effect on different NLG tasks. It reviews recent advances in prompting methods and their impact on NLG tasks, presenting prompt design as an input-level control mechanism that complements fine-tuning and decoding approaches. The paper introduces a taxonomy of prompting paradigms, outlines emerging trends and challenges, and proposes a framework that links design, optimization, and evaluation to support more controllable and generalizable NLG.


## 1 Introduction

The field of artificial intelligence (AI) has advanced significantly with the introduction of LLMs over the last few years. These LLMs have attained unprecedented performance in several downstream Natural Language Processing (NLP) tasks (Vatsal and Dubey, 2024), including, but not limited to, question answering, story telling, summarization, machine translation and sentiment analysis. Within this broader NLP landscape, Natural Language Generation (NLG) poses its own challenges. Despite the fluency and versatility of LLMs, generating high-quality text for diverse NLG tasks often demands careful guidance that goes beyond improvements in model architecture (Gatt and Krahmer, 2018; Pandey and Roy, 2023a) or task-specific fine-tuning (Xu et al., 2023). In this context, prompt engineering (Liu et al., 2023) has emerged as a promising paradigm for steering LLM outputs in NLG, enabling flexible control over style, structure and content, without additional retraining. Prompt engineering is, as shown in Figure 1, a technique whereby natural language instructions, or prompts, are used to guide an LLM's behavior responses, with the aim of improving accuracy, relevance and coherence in the generated output (Chen et al., 2025). This approach offers a practical, low-resource alternative for advancing NLG applications, making it increasingly relevant as the field moves towards building robust and adaptable generation systems.

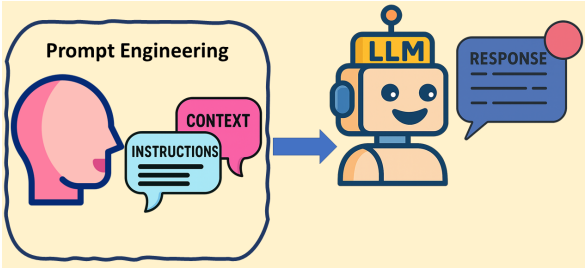


Figure 1: Graphical flow of how prompt engineering works: context of the conversation along with the instructions/prompts formulates input which guides an LLM towards generating specific output.

Prompt design plays a critical role in shaping the structure and coherence of a response across a wide range of NLG tasks (Schulhoff et al., 2024). Owing to its use of instructions and examples written in natural language, it constitutes a bridge between users and LLMs, letting users decide and guide an LLM's behavior. Despite its increasing popularity and significance, prompt engineering remains underrepresented in existing surveys on NLG. Most of the surveys focus on

innovation in architecture designs (Pandey and Roy, 2023b; Zarrieß et al., 2021), evaluation methods (Stent et al., 2005; Sai et al., 2022; Celikyilmaz et al., 2020), or the categorization of downstream tasks (Dong et al., 2022; Gatt and Krahmer, 2018; Santhanam and Shaikh, 2019). As a result, there is a need for a survey focusing just on prompting for NLG and its applications.

Thus, this survey offers a comprehensive synthesis of prompt engineering as an emerging paradigm for controllable NLG. It contrasts prompt-based control with fine-tuning and decoding strategies, introduces a taxonomy of prompting paradigms, and analyzes how prompts influence content, structure, and style without retraining. The paper also reviews evaluation methods, robustness issues, and the effectiveness of prompt-driven control across NLG tasks. Finally, it outlines emerging trends and proposes a systematic framework that formalizes prompt engineering as an input-level control approach for future research in prompt-based generation. Collectively, this work formalizes prompt engineering as an input-level control framework and establishes a conceptual foundation for future advances in prompt-based generation.

## 2 Comparison with Fine-Tuning and Decoding-Level Control

Prompt-based control operates at the input level, requiring no extra training and providing fast adaptability to new control goals. In contrast, fine-tuning allows deeper integration of control signals by aligning an LLM's internal representations with desired outputs. This achieves higher consistency than prompt-based NLG outputs, but at higher computational and data costs (Shin et al., 2025).

Decoding-level control (e.g., constrained beam search, nucleus sampling adjustments) (Naseh et al., 2023) manipulates the generation process after an LLM's next-token probabilities are computed, enabling some lexical or length constraints, but with limited flexibility for high-level aspects of generation, such as discourse structure, tone, or content framing (Holtzman et al., 2020).

Thus, prompt engineering holds a unique middle position. It is more cost-effective and adaptable than fine-tuning, while offering broader control dimensions than decoding-level control. This makes prompt engineering an effective strategy across various NLG tasks such as summarization, story generation, dialogue generation etc. (Vatsal and Dubey, 2024).

Empirical findings in recent studies both substantiate the functional distinctions between prompting, fine-tuning, and decoding-level methods, and expose a growing convergence through hybrid approaches that integrate elements of each. Parameter-efficient methods such as prefix tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) demonstrate that lightweight fine-tuning can approach the performance of full model adaptation, offering a middle ground between static prompts and full retraining. Similarly, decoding-level techniques such as NeuroLogic Decoding (Lu et al., 2020), contrastive search (Su and Collier, 2022), and contrastive decoding (Shi et al., 2024) enhance coherence, factual grounding, and stylistic alignment beyond conventional constrained beam search. At the intersection of these paradigms, interdisciplinary approaches such as task vectors (Belanec et al., 2024) and task arithmetic (Ilharco et al., 2023) combine the adaptability of prompting with the representational control of fine-tuning, enabling efficient model editing without full retraining.

Taken together, this suggests that future NLG systems may adopt *modular, compositional control strategies*, where prompts provide rapid adaptability, fine-tuning ensures deep alignment, and decoding mechanisms enforce reliability, with hybrid approaches offering flexible trade-offs depending on the task and resource profile.

## 3 Taxonomy of Prompting Techniques

In this section, we present a taxonomy of prompting paradigms and how they support different NLG tasks. We distinguish between ***foundational paradigms***, which provide the basis for prompt-based control, ***contextual paradigms***, that focus on adapting prompt strategies to distinct NLG tasks and use cases, and ***advanced reasoning paradigms***, which extend prompting with search, planning, and external actions. We also outline how well-suited these categories are for different NLG tasks (Table 1).

### 3.1 Foundational Paradigms

**Zero-shot Prompting (Radford et al., 2019).** This technique relies on carefully-curated prompts to guide the LLMs in performing specific NLG tasks, such as machine translation (MT) and story telling. Its strength lies in the quick deployment of

an idea with a relatively low design effort.

**Few-shot Prompting (Brown et al., 2020).** This technique leverages the idea of in-context learning to provide a few input-output pairs to improve an LLM's understanding of a given task. Eliciting even a few high-quality examples has yielded performance gains on different NLG tasks, such as dialogue generation and machine translation, steering the output towards certain stylistic nuances.

**Chain-of-thought (CoT) Prompting (Wei et al., 2022).** This technique takes its inspiration from how people decompose a complex task into smaller sub-tasks before arriving at the final solution. Along the same lines, this prompting paradigm provides instructions to an LLM in such a way that encourages a step-by-step, coherent reasoning process. CoT can help LLMs plan outlines, sub-points and narrative arcs for NLG tasks such as storytelling, report generation and summarization, thereby improving global coherence.

**Role Prompting (Kong et al., 2023).** This technique involves assigning a role to the LLM to enhance its understanding of the task. For example, if the model is prompted to act as a mathematician, it is likely to provide a correct step-by-step explanation of a mathematical concept (Van Buren, 2023). It also serves as an effective implicit CoT trigger, explaining its enhancements in reasoning capabilities.

### 3.2 Contextual Paradigms

**Thread-of-thought (ThoT) Prompting (Zhou et al., 2023).** This technique, which draws its inspiration from human cognitive processes, is designed to enhance the reasoning abilities of LLM models by asking them to select pertinent information from context comprising information from diverse sources, including user queries, conversation history, and external knowledge bases. ThoT has yielded substantial performance gains on tasks like conversational question answering and dialogue system where the generation of contextually appropriate answer is critical.

**Chain-of-event (CoE) Prompting (Bao et al., 2024).** This technique, which was proposed for summarization, consists of four steps: (1) extract specific events, (2) analyze and generalize the extracted events in more refined and concise form, (3) filter the generalized events to retain only those that cover most of the text, and (4) integrate the events selected in step (3) based on their chronological order or level of importance.

### 3.3 Advanced Reasoning Paradigms

**Program-of-Thought (Chen et al., 2023).** Built up on CoT, this approach represents reasoning as explicit program-like structures, improving interpretability and consistency in structured NLG tasks such as data-to-text reporting, contextual question-answering and conversational contextual question-answering tasks.

**Tree-of-Thoughts (ToT) (Yao et al., 2023; Long, 2023).** ToT expands CoT by exploring multiple reasoning paths as a search tree, selecting the best branch for the final generation. This enhance prompting capabilities for complex tasks requiring exploration and look-ahead reasoning. This structure allows language models to deliberately reason by assessing progress generated by thoughts in solving the problem. This paradigm benefits planning-heavy NLG such as story generation, report writing, and multi-step summarization.

**Self-Consistency Prompting (Wang et al., 2023).** Instead of relying on a single reasoning path, self-consistency samples multiple reasoning chains and aggregates the outputs. Self-Consistency utilizes a novel decoding strategy unlike the greedy one being used by CoT and consists of three key steps. The first step requires prompting the LLM using CoT, the second step samples diverse reasoning paths from LLM's decoder and the final step involves choosing the most consistent answer across multiple reasoning paths. This improves robustness in reasoning-intensive NLG tasks such as argument generation or complex summarization.

## 4 Prompting for Controlled NLG

Despite the LLMs ability to generate fluent and grammatically sound texts, other control dimensions are important in real-world applications, e.g., style, content, length and structure, which might not be achievable if we allow LLMs to generate text freely (Ajwani et al., 2024). Controlling model outputs to fit a set of constraints is the purview of controllable text generation (Lu et al., 2023), and prompting serves as a *'control lever'* for steering the outputs of LLMs, without requiring extensive parameter fine-tuning.

Table 1: Comparative Overview of Prompting Paradigms in NLG.

| Prompting Paradigm | Primary Strategy | Strengths | Limitations | Typical NLG Tasks |
|---|---|---|---|---|
| **Foundational Paradigms** | | | | |
| Zero-shot (Radford et al., 2019) | No examples; relies on pretrained generalization | Minimal setup cost, quick deployment | Highly sensitive to prompt phrasing | Classification, QA, machine translation |
| Few-shot (Brown et al., 2020) | In-context examples demonstrate task pattern | Improves task-specific performance | Token budget grows with examples; brittle to formatting | Summarization, dialogue generation, structured text |
| Chain-of-Thought (Wei et al., 2022) | Step-by-step reasoning in prompts | Improves reasoning transparency and coherence | Verbose; task-dependent gains | Explainable generation, QA, structured reports |
| Role Prompting (Kong et al., 2023) | Assigns persona or role to guide behavior | Boosts creativity, diversity, and framing | Relies on accurate persona crafting | Dialogue agents, creative writing, tutoring |
| **Contextual Paradigms** | | | | |
| Thread-of-Thought (Zhou et al., 2023) | Maintains discourse flow across turns | Enhances context tracking in long dialogues | Hard to automate; needs clear structure | Conversational QA, data-to-text NLG |
| Chain-of-Event (Bao et al., 2024) | Extracts and compresses event chains | Improves summary coherence and fluency | Narrow scope; task-specific design | Multi-document summarization |
| **Advanced Reasoning Paradigms** | | | | |
| Program-of-Thought (Chen et al., 2023) | Converts reasoning into program-like steps | Improves interpretability and structured reasoning | Requires task formalization | Data-to-text generation, structured reporting |
| Tree-of-Thoughts (Yao et al., 2023; Long, 2023) | Explores reasoning as a search tree | Supports planning and complex reasoning | Computationally expensive | Story generation, multi-step summarization, planning-heavy NLG |
| Self-Consistency (Wang et al., 2023) | Aggregates multiple reasoning chains | Improves robustness and factual reliability | Costly; may over-generate redundant outputs | Argument generation, complex summarization |

### 4.1 Content Control: Topical Constraints and Lexical Anchoring

The limits of generative LLMs are unclear, yet from research perspective, we must be able to determine what makes them successful and what causes them to fail (Lu et al., 2023). Carefully crafted prompts can enforce topical constraints by explicitly specifying the domain or keywords to be included in the output. For example, " Generate a short explanation about the challenges blind and low vision individuals face in accessing data visualizations in educational settings" can anchor the generation of text around the relevant topic and subtopics. Similarly, lexical anchoring, which refers to specifying important information to the LLM before reaching a decision, can guide the model to generate domain-specific information without re-training (Tian and Zhang, 2024). For example, prompting with "Write a paragraph about the challenges in web accessibility for blind users. Make sure to include the terms *tactile graphics*, *screen reader*, and *access barrier*" anchors the model in accessibility terminology and context. These approaches are useful for NLG tasks such as question answering (Zhu et al., 2021), story telling (Fan et al., 2018), and summarization (Xu et al., 2024) etc.

### 4.2 Structure Control: Length and Discourse Organization

Users often expect generated texts to fall within a specific length range, making length controlled generation an important topic (Jie et al., 2024). For pre-trained language models, the most widely applied technique for length control is prompt-based fine-tuning (e.g., 'summarize in two sentences' or 'provide a bullet-point list of advantages of living in coastal areas) (Liu et al., 2023). In addition, prompts can also guide an LLM to simulate the discourse structure of human-written text (Ghazvininejad et al., 2022).

### 4.3 Style Control: Formality, Emotion, and Tone

Prompting can effectively condition the formality level ('write in a formal academic style' vs. 'explain in a casual, friendly tone'), emotional coloring ('write a comforting message to someone anxious about exams'), and persona-based tone ('as a supervisor, advise on thesis writing'). Drawing its inspiration from the impact of language on human

performance, EmotionPrompt (Li et al., 2023) utilizes emotional cues to help improve an LLM's emotional appeal. Compared to style transfer methods requiring paired data or explicit attribute modeling, prompting offers a flexible and low-resource alternative (Yang and Carpuat, 2025). In addition to written output, the emotional prompts may be used to control and guide the emotional tone of synthesized speech (Bott et al., 2024).

While prompting is effective for controlling individual stylistic dimensions such as formality or tone, achieving *fine-grained multi-attribute control* (e.g., simultaneously adjusting formality, sentiment, and persona) remains a significant challenge. Prompt formulations often lead to conflicts between attributes, resulting in outputs that privilege one dimension over another. Recent decoding-time strategies, such as contrastive decoding (Shi et al., 2024), highlight the importance of leveraging context-aware constraints to balance competing stylistic goals. However, extending such approaches to multi-attribute style control via prompting alone remains underexplored and may require hybrid solutions that combine manual prompt design, optimization-based methods, and decoding-level interventions.

## 5 Evaluation and Prompt Robustness

To systematically understand the impact of prompt engineering, we need to examine how different evaluation metrics capture output quality. Unlike standard NLG, prompted NLG introduces unique challenges: performance is often more sensitive to phrasing, reasoning depth, and context handling. As such, traditional metrics may not fully reflect the effectiveness of prompt-based control.

Both ***human-centered*** and ***automatic*** metrics have traditionally been used to assess NLG quality, each with distinct advantages and limitations. More recently, a third paradigm, known as ***LLM-as-a-Judge***, has emerged, leveraging LLMs themselves as evaluators to complement human and automatic methods.

**Human Evaluation.** Human evaluation relies on expert or crowd annotators to assess dimensions that are difficult to quantify automatically, such as coherence, fluency, factuality, and stylistic appropriateness (Holtzman et al., 2020). It is increasingly used in abstract tasks like summarization and creative writing (Stiennon et al., 2020; Yao et al., 2023; Wang et al., 2024). However, human evaluation is costly, time-consuming, and may suffer from annotator bias or inconsistency, making large-scale benchmarking difficult.

**Automatic Evaluation.** Automatic metrics are fast and scalable, providing quantitative assessments of different prompting strategies. Classical overlap-based metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) remain widely used, while embedding-based metrics such as BERTScore (Zhang et al., 2019) and MoverScore offer higher semantic alignment. More recent approaches include *contrastive decoding* methods (Shi et al., 2024), which evaluate preference between candidate generations, and specialized metrics for safety, such as toxicity and bias scores (Gehman et al., 2020). Despite their efficiency, automatic metrics often fail to capture nuanced qualities like creativity, discourse structure, or reasoning validity (Sai et al., 2022).

**LLM-as-a-Judge.** A recent trend is to use LLMs themselves as evaluators. Here, an LLM is prompted to rate outputs according to quality dimensions such as coherence, faithfulness, or style. Studies show that GPT-4 and similar models correlate well with expert human judgments, enabling scalable and cost-efficient evaluation (Zheng et al., 2023). However, this approach raises concerns that LLM evaluators may inherit the biases of the model being judged, may be vulnerable to prompt hacks, and risk circularity when the same model family is used for both generation and evaluation (Gu et al., 2024). Developing reliable protocols for "LLM-as-a-judge" remains an open challenge.

**Discussion.** Evaluating prompted NLG thus remains difficult for several reasons:

- Sensitivity to minor prompt variations complicates reproducibility of evaluation results (Sclar et al., 2023);
- Existing metrics underrepresent attributes like reasoning faithfulness and contextual coherence (Li et al., 2024);
- LLM-based judges, while promising, raise concerns about bias propagation and lack of interpretability (Novikova et al., 2025).

A comprehensive evaluation framework should therefore triangulate across human, automatic, and LLM-based metrics to capture the full spectrum of

quality (Pan et al., 2024), especially as prompting becomes central to output-level control.

### 5.1 Prompt Sensitivity and Brittleness, and Optimization

LLM-generated outputs can be highly sensitive and variable to the prompt phrasing and structure, with performance often differing markedly across models and tasks based on these nuances (Zhuo et al., 2024; Chatterjee et al., 2024; Salinas and Morstatter, 2024). This sensitivity manifests in two formats: i) *local sensitivity* refers to the small lexical changes such as synonyms or reordering of instructions may lead to noticeable difference in tone, quality and relevance of the generated response. ii). *global brittleness* refers to changes in prompt length or specificity can lead to failures in adhering to the intended task, indicating a lack of robustness in prompt-based control compared to more structured fine-tuning methods.

To mitigate these issues, the field of ***prompt optimization*** has emerged. Instead of relying solely on manual trial-and-error, optimization methods learn or refine prompts in a systematic manner. *Prompt tuning* (Lester et al., 2021) introduces soft prompt vectors learned while keeping model parameters frozen, offering consistency across tasks. *Prefix tuning* (Li and Liang, 2021) similarly prepends trainable vectors to the input sequence, enabling parameter-efficient adaptation. Other approaches such as *task vectors* and *task arithmetic* (Belanec et al., 2024; Ilharco et al., 2023) allow compositional editing of model behavior by adding or subtracting task-specific directions in embedding space. Automated prompt search methods, including gradient-based and reinforcement-learning approaches (Shin et al., 2020; Cheng et al., 2024), further extend this line by discovering effective discrete prompts without human intervention.

While these optimization strategies enhance robustness and portability, they trade off interpretability and creative flexibility (Patel et al., 2025). Manual prompt design remains attractive for low-resource, domain-specific tasks, whereas optimization-based methods provide scalability, reproducibility, and systematic control. A comprehensive framework for when to employ manual design, optimization, or hybrid approaches remains an important open research question.

## 6 Emerging Trends and Open Challenges

The field of prompt engineering is evolving through various key trends. *Prompting with retrieval augmented generation* (RAG) (Lewis et al., 2020) enforce factual grounding and reduce hallucinations by pairing text generation with retrieval. *Prompt engineering as programming* (Sharma et al., 2024) is capturing attentions via prompt DSL, a domain-specific language for refining prompts and monitoring the inputs to the LLM-based chatbots. More recently, the concept of *context engineering* (Mei et al., 2025) has been introduced, extending prompt design into a broader process of selecting, organizing, and weighting relevant contextual information before it is provided to the model. This approach views prompts not as standalone inputs but as components of a larger pipeline that integrates retrieval, augmentation, and orchestration strategies. In doing so, it connects prompt engineering with broader system-level design of LLM applications. Another emerging trend is the application of prompt engineering in *multilingual NLG* (Vatsal et al., 2025). Recent studies shows that prompts can enable cross-lingual transfer and adaptation, allowing LLMs to generalize across diverse linguistic and cultural contexts. However, multilingual prompting also highlights new opportunities for advancing context-aware and culturally grounded design, where lexical choices, stylistic framing, and contextual cues are adapted to language-specific conventions rather than translated directly from English.

Despite its promise, prompt engineering faces several persistent challenges that limit its reliability and generalizability.

### 6.1 Prompt Brittleness

LLM outputs are highly sensitive to small variations in prompt wording, ordering, or formatting. Even minor lexical substitutions can produce drastically different generations, leading to instability in downstream applications (Salinas and Morstatter, 2024; Chatterjee et al., 2024; Zhuo et al., 2024). This brittleness raises concerns for reproducibility and makes it difficult to establish robust prompt design principles.

### 6.2 Bias Amplification

Prompts can inadvertently reinforce or even amplify social biases embedded in pre-trained LLMs. For example, toxicity benchmarks have shown that

prompts conditioned on specific demographics can trigger harmful stereotypes (Gehman et al., 2020). While lexical anchoring and prompt tuning may mitigate these risks, bias amplification remains a core obstacle for deploying prompt-engineered systems responsibly.

### 6.3 Factual Accuracy and Hallucination

Although prompt engineering can steer style and content, it often fails to ensure factual accuracy. LLMs are prone to hallucinating details or presenting plausible-sounding but incorrect information. Retrieval-augmented prompting (Lewis et al., 2020) has shown promise, but ensuring factual reliability across domains remains unresolved.

### 6.4 Evaluation Gaps

Evaluating prompts is challenging. Human evaluations provide nuanced insights but are costly and inconsistent, while automatic metrics (e.g., BLEU, ROUGE, BERTScore) capture only surface similarity. Recent trends in using LLMs as judges offer scalability but introduce new risks of model bias (Sai et al., 2022; Celikyilmaz et al., 2020). Without standardized benchmarks, it is difficult to compare techniques fairly.

### 6.5 Generalization and Portability

Prompts designed for one domain, language, or model architecture often fail to transfer effectively (Perez et al., 2021). This lack of portability is particularly problematic in multilingual and cross-domain NLG, where prompts must adapt to diverse contexts without extensive re-engineering.

## 7 Toward a Systematic Framework for Prompt Engineering

While existing research highlights the fragility and limited portability of prompts across domains, languages, and model architectures (Perez et al., 2021), the field still lacks a unified framework that connects design principles, optimization strategies, and evaluation practices. Establishing such a framework is critical for advancing prompt engineering from an empirical craft to a principled science of input-level control.

We propose that progress in prompt engineering can be structured around three interdependent dimensions namely: ***i. design***, ***ii. optimization***, and ***iii. evaluation***. Each dimension captures a complementary aspect of how prompts are constructed, adapted, and validated. *Design* focuses on abstraction and reusability; *optimization* emphasizes robustness and scalability; and *evaluation* ensures reliability, fairness, and transferability across tasks and languages. Table 2 summarizes these dimensions, the methods that operationalize them, and the trade-offs that arise when balancing interpretability, flexibility, and control.

### 7.1 Design Principles

Prompt design should prioritize abstraction over specificity. Reusable templates, role-based prompting, and culturally neutral phrasing facilitate cross-domain and multilingual portability (Vatsal and Dubey, 2024; Vatsal et al., 2025). At the same time, excessive abstraction may reduce expressiveness or limit stylistic nuance, underscoring the need for modular design that retains both flexibility and generality.

### 7.2 Optimization

Portability often depends on lightweight adaptation mechanisms that refine prompts without retraining the entire model. Methods such as discrete prompt search (Shin et al., 2020), Prefix and Prompt Tuning (Li and Liang, 2021; Lester et al., 2021), and decoding-level interventions like contrastive decoding (Shi et al., 2024) exemplify scalable solutions for aligning prompts across architectures. Yet these optimization-based strategies frequently trade interpretability and creative flexibility for robustness, raising open questions about how to balance manual and automated prompt design (Patel et al., 2025).

### 7.3 Evaluation

A systematic framework also requires rigorous stress-testing across linguistic, cultural, and domain boundaries. Complementary metrics, BLEU, ROUGE, BERTScore, quantify task performance, while safety-oriented scores such as toxicity and bias (Gehman et al., 2020) and emerging model-based evaluators (LLM-as-a-Judge) (Zheng et al., 2023; Gu et al., 2024) assess quality and fairness. Ensuring the transparency and reproducibility of such evaluations remains a central research priority.

## 8 Conclusion

Prompting has evolved into a core technique for LLM-driven NLG, enabling controlled, efficient generation without retraining. However,

Table 2: Core dimensions of a systematic framework for prompt engineering.

| Dimension | Core Practices | Illustrative Methods / References | Trade-offs |
|---|---|---|---|
| **Design** | Abstraction over specificity; modular, reusable prompt templates | Role-based and task-agnostic prompting (Vatsal and Dubey, 2024); multilingual and culturally neutral phrasing (Vatsal et al., 2025) | May reduce creativity or cultural nuance |
| **Optimization** | Lightweight adaptation; search-based or continuous tuning | Discrete prompt search (AutoPrompt) (Shin et al., 2020); Prefix Tuning (Li and Liang, 2021); Prompt Tuning (Lester et al., 2021); Contrastive Decoding (Shi et al., 2024) | Reduced interpretability; limited creative flexibility |
| **Evaluation** | Systematic stress-testing across domains and languages | BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019); Bias and Toxicity scores (Gehman et al., 2020); LLM-as-a-Judge (Zheng et al., 2023) | Metric bias; reproducibility and transparency challenges |

prompt design often involves repetitive and time-consuming debugging, as small phrasing changes can lead to unpredictable outputs.

To advance the field, prompt engineering should be formalized with frameworks, benchmarks and theory that support robust, scalable, and reusable prompt design for the next generation of NLG systems. There is also a pressing need for evaluation metrics that more accurately reflect the effectiveness and control capabilities of prompts.

## 9 Limitations

Despite offering a focused overview of prompt engineering for NLG, this survey has several limitations. Findings discussed are largely conceptual and may not generalize across domains, languages, or LLM architectures without further empirical validation. While we highlight issues such as prompt sensitivity and evaluation gaps, practical mitigation strategies and theoretical modeling could not be explored in depth. Future work is needed to extend the taxonomy, assess generalizability across multilingual settings, and ground prompt engineering in more rigorous empirical and theoretical frameworks.

## References


Rohan Deepak Ajwani, Zining Zhu, Jonathan Rose, and Frank Rudzicz. 2024. Plug and play with prompts: A prompt tuning approach for controlling text generation. *arXiv preprint arXiv:2404.05143*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Songlin Bao, Tiantian Li, and Bin Cao. 2024. Chain-of-event prompting for multi-document summarization by large language models. *International Journal of Web Information Systems*, (ahead-of-print).

Róbert Belanec, Simon Ostermann, Ivan Srba, and Mária Bieliková. 2024. Task prompt vectors: Effective initialization through multi-task soft-prompt transfer. *CoRR*, abs/2408.01119.

Thomas Bott, Florian Lux, and Ngoc Thang Vu. 2024. Controlling emotion in text-to-speech with natural language prompts. *arXiv preprint arXiv:2406.06406*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2025. Unleashing the potential of prompt engineering for large language models. *Patterns*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Trans. Mach. Learn. Res.*, 2023.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Black-box prompt optimization: Aligning large language models without model training. In

*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3201–3219. Association for Computational Linguistics.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2022. A survey of natural language generation. *ACM Computing Survey*, 55(8).

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369.

Marjan Ghazvininejad, Vladimir Karpukhin, Vera Gor, and Asli Celikyilmaz. 2022. Discourse-aware soft prompting for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4570–4589.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Renlong Jie, Xiaojun Meng, Lifeng Shang, Xin Jiang, and Qun Liu. 2024. Prompt-based length controlled generation with multiple control types. *arXiv preprint arXiv:2406.10278*.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2023. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023. Large language models understand and can be enhanced by emotional stimuli. *arXiv preprint arXiv:2307.11760*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Jieyi Long. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Albert Lu, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2023. Bounding the capabilities of large language models in open text generation with prompt constraints. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1982–2008.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. 2025. A survey of context engineering for large language models. *CoRR*, abs/2507.13334.

Ali Naseh, Kalpesh Krishna, Mohit Iyyer, and Amir Houmansadr. 2023. Stealing the decoding algorithms of language models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1835–1849.

Jekaterina Novikova, Carol Anderson, Borhane Blili-Hamelin, Domenic Rosati, and Subhabrata Majumdar. 2025. Consistency in language models: Current landscape, challenges, and future directions. *arXiv preprint arXiv:2505.00268*.

Qian Pan, Zahra Ashktorab, Michael Desmond, Martín Santillán Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. 2024. Human-centered design recommendations for LLM-as-a-judge. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pages 16–29. ACL.

Abhishek Kumar Pandey and Sanjiban Sekhar Roy. 2023a. Natural language generation using sequential models: A survey. *Neural Process. Lett.*, 55(6):7709–7742.

Abhishek Kumar Pandey and Sanjiban Sekhar Roy. 2023b. Natural language generation using sequential models: A survey. *Neural Processing Letters*, 55(6):7709–7742.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 311–318.

Oam Patel, Jason Wang, Nikhil Shivakumar Nayak, Suraj Srinivas, and Himabindu Lakkaraju. 2025. Towards interpretable soft prompts. *arXiv preprint arXiv:2504.02144*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Computing Survey*, 55(2).

Abel Salinas and Fred Morstatter. 2024. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4629–4651.

Sashank Santhanam and Samira Shaikh. 2019. A survey of natural language generation techniques with a focus on dialogue systems-past, present and future directions. *arXiv preprint arXiv:1906.00500*.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Reshabh K Sharma, Vinayak Gupta, and Dan Grossman. 2024. SPML: A dsl for defending language models against prompt attacks. *arXiv preprint arXiv:2402.11755*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791.

Jiho Shin, Clark Tang, Tahmineh Mohati, Maleknaz Nayebi, Song Wang, and Hadi Hemmati. 2025. Prompt engineering or fine-tuning: An empirical assessment of llms for code. In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 490–502. IEEE.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. page 341–351.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *arXiv preprint arXiv:2210.14140*.

Yuan Tian and Tianyi Zhang. 2024. Selective prompt anchoring for code generation. *CoRR*, abs/2408.09121.

David Van Buren. 2023. Guided scenarios with simulated expert personae: a remarkable strategy to perform cognitive work. *arXiv preprint arXiv:2306.03104*.

Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different NLP tasks. *CoRR*, abs/2407.12994.

Shubham Vatsal, Harsh Dubey, and Aditi Singh. 2025. Multilingual prompt engineering in large language models: A survey across nlp tasks. *arXiv preprint arXiv:2505.11665*.

Rui Wang, Hongru Wang, Fei Mi, Boyang Xue, Yi Chen, Kam-Fai Wong, and Ruifeng Xu. 2024. Enhancing large language models against inductive instructions with dual-critique prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5345–5363.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Lei Xu, Mohammed Asad Karim, Saket Dingliwal, and Aparna Elangovan. 2024. Salient information prompting to steer content in prompt-based abstractive summarization. *arXiv preprint arXiv:2410.02741*.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Xinchen Yang and Marine Carpuat. 2025. Steering large language models with register analysis for arbitrary style transfer. *arXiv preprint arXiv:2505.00679*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Sina Zarrieß, Henrik Voigt, and Simeon Schüz. 2021. Decoding methods in neural language generation: a survey. *Information*, 12(9):355.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. 2023. Thread of thought unraveling chaotic contexts. *arXiv preprint arXiv:2311.08734*.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *CoRR*, abs/2101.00774.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1950–1976.