Feedback-Aware Prompt Optimization Framework for Generating Job Postings

Suraj Maharjan and Ainur Yessenalina and Srinivasan H. Sengamedu
Amazon.com, Inc.
{mhjsuraj, yessenal, sengamed}@amazon.com

Abstract

Job postings are critical for recruitment, yet large enterprises struggle with standardization and consistency, requiring significant time and effort from hiring managers and recruiters. We present a feedback-aware prompt optimization framework that automates high-quality job posting generation through iterative human-in-the-loop refinement. Our system integrates multiple data sources: job metadata, competencies, organization's compliance guidelines, and organization brand statement, while incorporating human feedback to continuously improve prompt quality through multi-LLM validation. We evaluate our approach using LLM-as-a-judge on 1,056 job postings while also performing human evaluation on a smaller subset across three dimensions: Standardization, Compliance, and User Perception. Our results demonstrate high compliance rates and strong satisfaction scores in both automated and human evaluation, validating the effectiveness of our feedback-aware approach for enterprise job posting generation.

1 Introduction

In today's highly competitive job market, crafting effective job descriptions is crucial for attracting qualified candidates and ensuring successful recruitment outcomes. Job postings serve as the critical interface between organizations and potential talent, directly influencing both the quality and efficiency of recruitment outcomes. Particularly in large enterprises, where hundreds or thousands of positions are filled annually across diverse job families and levels, the quality and consistency of job descriptions become paramount. Beyond merely attracting candidates, job postings form the foundation for effective talent matching, enabling organizations to identify exceptional candidates and align them with appropriate roles. However, even the effectiveness of sophisticated candidate-job matching systems can only be as strong as the quality of the underlying job descriptions that they ultimately depend upon.

Despite their importance, creating high-quality job postings presents significant challenges in enterprise settings. Writing job descriptions is not only time-consuming, but the absence of standardization across job families, coupled with limited access to templates and writing assistance tools, places a substantial burden on hiring managers and recruiters. These complexities require significant time and effort to draft effective job descriptions that are both accurate and effective in attracting top talent.

This challenge is further compounded by several systemic issues that impact recruitment quality and efficiency. First, job-specific competency requirements are not always fully defined at the initial job creation stage. For efficiency, hiring managers may adapt existing job postings with modifications to create new openings. While this approach saves time, it can sometimes result in job descriptions that do not fully reflect current role requirements or evolving organizational needs, potentially limiting the ability to attract the most qualified candidates. Furthermore, many hiring managers, in the absence of proper guidance, struggle to craft compelling job descriptions, resulting in postings that fail to communicate the role's value proposition or growth opportunities to top-tier candidates. This problem is exacerbated by vague job descriptions that leave job seekers uncertain about their fit for the position, leading to either under-application from qualified candidates or over-application from misaligned ones. Such ambiguity also impairs recruiter effectiveness, as unclear requirements make it difficult to efficiently screen and prioritize candidates.

These challenges highlight the urgent need for an automated, scalable solution that can generate high-quality, compliant, and standardized job postings while reducing the burden on hiring managers and recruiters. In this paper, we present a feedback-

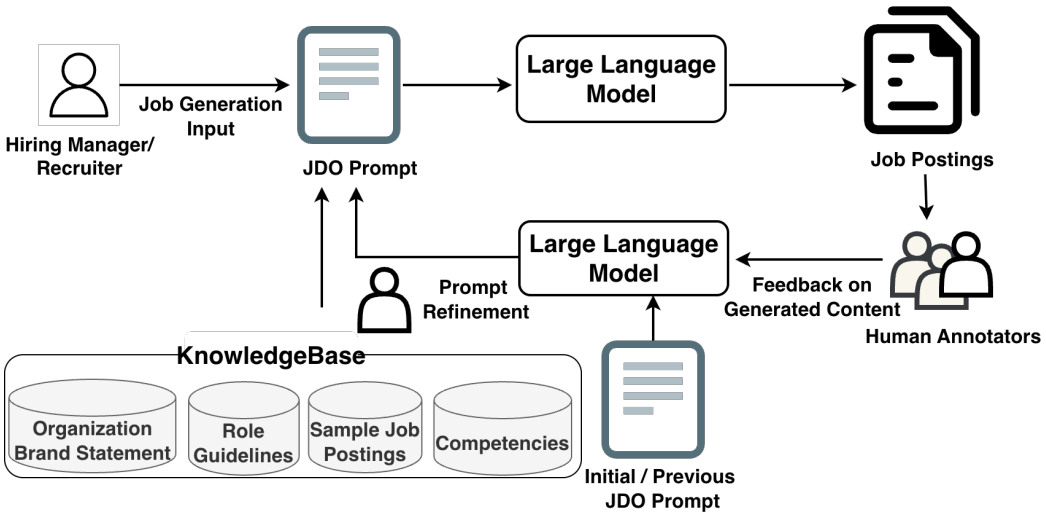

Figure 1: Feedback-aware prompt refinement system.

aware prompt optimization framework that uses iterative human-in-the-loop feedback mechanisms to refine the prompt for generating job postings. We collect human feedback on generated job postings and use it to instruct LLMs to refine the prompt. Our approach combines organizational knowledge and compliance requirements to produce high-quality job descriptions that are simultaneously compliant, engaging, and effective for talent attraction and matching.

2 Methodology

Figure 1 illustrates the overall system design for the job posting generation with the prompt refinement process. Our system implements a feedback-aware prompt optimization framework that integrates multiple data sources, including job metadata, organizational compliance guidelines, role-specific guidelines, job family and level specific functional and core competencies, organization brand statement, and recently published job postings (Brown et al., 2020). The framework employs an iterative refinement process where human annotators review generated job postings and provide feedback that guide LLMs in improving the generation prompt. This human-in-the-loop architecture enables continuous prompt adaptation based on the quality assessments and the expert feedback.

As shown in Figure 1, we first develop an initial prompt containing instructions on organizational compliance guidelines, which we then refine. We enrich the prompt with organizational role guidelines, job family and level competencies, brand values, and a few examples of recently posted job postings. We then obtain specific inputs from hiring managers and combine them with the enriched prompt to generate job postings. These job postings are manually evaluated and annotated by subject matter experts. We gather all the feedback provided by them and then instruct the LLM to update the instructions by taking into account the new feedback. We then test the prompt to ensure the changes are effective. After refining the generation prompt through this iterative process, we use it in two architectural approaches for job posting generation: Single-Prompt and Multi-Agent.

Single-Prompt Approach: This approach makes a single LLM call using the refined prompt along with comprehensive input data. The inputs include job metadata (title, level, job family, and organization) and hiring managers/recruiters provided responses to five key questions addressing: (1) required skills and competencies, (2) long-term growth opportunities, (3) role differentiation within the organization, (4) team context and culture, and (5) typical day-to-day responsibilities. The prompt incorporates role-specific guidelines, functional and core competencies, organization brand statements, and representative job posting examples from the past six months. All generated content adheres to the organization's job description standards and compliance requirements. The average prompt word count is 4866.48 ± 427.87.

Multi-Agent Approach: This approach employs a two-stage architecture consisting of a Writer Agent and a Reviewer Agent. The Writer Agent generates an initial job posting draft using the same refined prompt and enriched input data as the Single-

Prompt approach. The Reviewer Agent then executes multiple specialized review tasks, each focusing on a specific section: job title, description, key responsibilities, a-day-in-the-life, about team, and additional information for internal candidates. Each review task performs compliance verification for its designated section and generates revisions as needed (Shinn et al., 2023). Finally, a formatting task aggregates all reviewed and revised sections, combining them into a properly structured markdown document.

3 Dataset

We sampled 1,056 job postings from nine job families spanning four organizational levels within corporate roles. The levels range from Level 4, representing entry-level positions, to Level 7, corresponding to Principal and Senior Manager positions. For each job family and level, we targeted 30 job postings, achieving this goal for all categories except Support Engineering, where fewer senior-level postings were available. The sample comprises 82.1% Individual Contributor (IC) roles and 17.9% Managerial roles. Table 1 shows the overall data distribution across job families and levels. The nine job families were selected based on their hiring volume. This ensures that our analysis captures the majority of hiring patterns and requirements within the organization.

Job Family	Level 4	Level 5	Level 6	Level 7
Account Mgmt	30	30	30	30
Buying/Planning	30	30	30	30
Finance/Planning	30	30	30	30
Program Mgmt	30	30	30	30
Tech Sales	30	30	30	30
Software Dev	30	30	30	30
Solution Arch	30	30	30	30
Support Eng	30	30	30	6
Tech Prog Mgmt	30	30	30	30

Table 1: Job posting distribution across families and levels.

4 Experiments

We compare two generation approaches: a single-prompt approach and a multi-agent approach. To systematically evaluate the generated job postings, we develop a comprehensive evaluation framework covering three dimensions: compliance with organizational standards, standardization across role guidelines, and user perception/satisfaction. We conduct two phases of human evaluation and annotation to assess content quality and gather feedback, which we incorporate into iterative prompt refinement.

4.1 Experimental Setup

As described in Section 2, our system requires hiring managers and recruiters to provide the following key information: (1) top three required skills, (2) long-term expectations, (3) potential project details, (4) daily responsibilities, (5) team structure, and (6) additional information (optional). To evaluate our approach across diverse positions, we utilize a dataset of existing job postings spanning nine job families and multiple seniority levels, from entry-level to principal and senior management roles (see Section 3).

Our experimental pipeline consists of two stages. First, we prepare inputs from hiring managers and recruiters by extracting structured inputs from existing job postings using Claude 3.7 Sonnet with our engineered prompt. Second, we use these extracted inputs to generate new job postings by incorporating additional organizational context, including role-specific guidelines, job family and level-specific functional and core competencies, organization brand statements, and examples from recent postings. We evaluate two generation approaches: a single-prompt method and a multi-agent system. For all LLM operations, we configure the model with $temperature = 0.7$, $top\_p = 0.7$, and $top\_k = 100$ to balance output diversity and consistency.

4.2 Evaluation Framework

Our primary goal is to help hiring managers create better job postings by combining standardized templates with GenAI tools that capture each role's unique requirements. Thus, we evaluate the generated job postings across following three key dimensions:

Compliance: This dimension evaluates whether generated content adheres to organizational job description standards and compliance criteria. For instance, it checks for appropriate language and tone, bias-free content, and exclusion of internal team names, code names, or unexplained acronyms.

Standardization: This dimension evaluates whether the generated content maintains consistency within the same job family and level, and job postings exhibit uniform formatting and structure.

User Perception: This dimension evaluates whether users perceive the generated content to be better than content produced by the current process. Specifically, we assess whether users believe the content improves upon our current process in two critical ways: its ability to attract higher-quality candidates and its effectiveness in helping candidates self-select into roles that align with their skills.

Moreover, to evaluate the overall effectiveness/satisfaction of our system, we collected comprehensive feedback from participants regarding the generated job descriptions. Specifically, we assessed whether the content appropriately emphasized information relevant to job seekers, and measured participants' overall satisfaction with the quality of the generated content. We also investigated the perceived likelihood of adoption by examining how probable participants believed it would be for hiring managers and recruiters to utilize the tool in their actual job posting workflows. Finally, we solicited open-ended suggestions for system improvements to identify potential areas for future enhancement and refinement of the tool's capabilities.

We performed two rounds of human annotations and evaluations to gather feedback for prompt improvement and refine evaluation questions based on the aforementioned three key dimensions.

4.2.1 Phase 1: Human Evaluation

In Phase 1, we reached out to 15 internal team members to review the generated job postings. Each user was asked to review three job descriptions within one of three job families: Software Dev, Support Eng, and Program Mgmt and complete a survey.

We received 24 responses to compliance-related questions and 8 responses to standardization, user perception, and overall satisfaction questions. We obtained an average compliance score of 86.69%. The survey also revealed confusion regarding one question on whether the generated content included internal team names, internal code names, or team-specific acronyms. Our deep dive by two different team members confirmed that question might have been not conveyed clearly and lead to false positives. All respondents agreed that the generated content was consistent and aligned with the role guidelines. For user perception and overall satisfaction, we obtained an average score of 4.17 and 4.56 (on a 5-point scale), respectively.

We also conducted large-scale evaluation using LLM-as-a-judge framework with three models from different families (Claude Sonnet 4, Claude Sonnet 3.5 v2, and Nova Pro (Intelligence, 2024)) to mitigate model-specific biases. We used the same set of sampled 68 recent job postings spanning three job families (Software Development, Support Engineering, Program Management) across four levels (L4–L7). For each posting, we generated inputs from hiring managers and recruiters using Claude 3.7 Sonnet, then regenerated the job posting using our refined prompt with these inputs, job metadata, role guidelines, brand statements, competencies, and sample job postings from the corresponding job family and level.

Each LLM independently evaluated the generated postings across three dimensions: Compliance, Standardization, and User Perception, using structured prompts that incorporated the generated content and relevant contextual information. We aggregated scores from multiple LLMs by averaging Likert responses (1–5) and majority voting for binary questions. We obtained strong system performance: 97% average compliance, 100% standardization (all postings contained required sections with responsibilities matching role guidelines), and high user perception scores (4.43/5 for candidate attraction, 4.48/5 overall satisfaction). These scores also correlate with the human evaluation.

Phase 1 provided early learning to improve the prompt. At a high level, feedback on acronyms, standardizing the job title, mention of location, and other were helpful feedback to improve the prompt. We also changed the wording of the question about internal team names, internal code names or team-specific acronyms, and explained that it is acceptable to use them if explained in text.

4.2.2 Phase 2: Human Evaluation

Following prompt refinement based on Phase 1 feedback, we conducted Phase 2 human evaluation with 17 hiring managers and recruiters using the updated prompt. We followed the same evaluation methodology as Phase 1. However, we updated the questionnaire based on Phase 1 learning. Phase 2 evaluated newly sampled job postings from three job families (Software Development, Support Engineering, and Program Management) across all four job levels (L4–L7). We collected 68 responses for compliance-related questions and 16 responses for standardization, user perception, and overall satisfaction questions. Similar to Phase 1 results, the evaluation results demonstrate strong performance

across all dimensions. The results demonstrate strong performance: 81.37% average compliance score, 81.25% for standardization, and average ratings of 3.77 for user perception and 3.81 for overall satisfaction (on a 5-point Likert scale).

We also received valuable feedback from human annotators for prompt refinement. They emphasized ensuring gender-neutral language throughout, maintaining consistent structural templates per job family, and eliminating vague generalized statements such as "executing flawlessly" among other improvements. We incorporated these refinements into subsequent iterations to further improve the system prompt.

5 Results

Table 2 presents the relative improvement of the multi-agent system over the single prompt baseline for 1,056 generated job postings across nine different job families and levels. To scale the evaluation process, we employ an LLM-as-a-judge framework using three models (Claude Sonnet 4, Claude Sonnet 3.5 v2, and Nova Pro) from different model families. The idea is to mitigate potential biases that can be present when using a single model. The generated content is evaluated along three dimensions: Compliance (Yes/No), Standardization (Yes/No), and User Perception (1–5 scale). For numeric responses (1–5 scale), we compute the mean across all three LLMs whereas for binary responses (Yes/No), we use majority voting instead. We also ask the judge LLMs to provide an explanation before producing the final answer to the evaluation questions. Since latency is a critical factor for user experience in a production environment, we also evaluate on this dimension along with the aforementioned metrics.

The results show that the multi-agent system achieves better performance than the single-prompt approach across all quality metrics: compliance, standardization, user perception, and overall satisfaction, although the improvement is marginal in most metrics. However, this improvement comes at a significant computational cost - the multi-agent system has an approximately 7× higher latency due to the fact that it makes multiple sequential LLM calls to generate the final job posting. Even though we employ asynchronous calls for the compliance check and revision steps, the multi-agent system still incurs a significantly longer time in the generation of the final job posting.

Although the single-prompt approach is demonstrably faster, even for this approach, the latency exceeds acceptable thresholds for real-time deployment. To address this challenge, we implement a streaming architecture that progressively delivers content to the frontend as it is generated. This approach achieves a time-to-first-token (TTFT) well within acceptable limits and significantly improves perceived user experience by reducing the initial wait time to approximately 2.28 seconds. Given the minimal quality improvement (less than 1% across metrics) and the substantial 7× latency disadvantage of the multi-agent system, we recommend deploying the single-prompt approach with streaming enabled for production use. This configuration provides an optimal balance between content quality and user experience.

6 Related Work

Prompt Optimization: Several studies have proposed automatic and manual methods for optimizing prompts to improve the quality of LLM-generated outputs (Lu et al., 2022; Yuksekgonul et al., 2025; Khattab et al., 2024; Li and Klinger, 2025; Yan et al., 2025; Wang et al., 2025; Zhen et al., 2025). DSPy (Khattab et al., 2024) introduces a framework for automatic prompt optimization. TextGrad (Yuksekgonul et al., 2025) proposes automatic differentiation via text, treating prompts as differentiable parameters optimized using textual feedback as gradients. Lin et al. used human preference feedback to optimize the prompt for LLMs. Li and Klinger used interactive prompt optimization approach with human in the loop to optimize the prompt. Our work optimizes prompts by instructing an LLM with expert feedback collected on generated job postings through multiple rounds of evaluation.

Multi-Agent Systems: Multi-agent autonomous or semi-autonomous systems are being widely explored across multiple domains (Xiao et al., 2025; Du et al., 2024; Islam et al., 2024). Li et al. used four agents: generator agent, visual critique agent, code critique agent, and revision agent to generate the code to create the reference chart image. Similarly, Su et al. proposed LLM based multi agent system, Virtual Scientists (VIRSCI), to collaboratively generate, evaluate, and refine research ideas. Shao et al. used multi-agent collaboration to write Wikipedia-like articles. In this paper, we also explore multi-agent system for generating job post-

Metrics	Single Prompt	Multi-Agent
Compliance (Y/N)	-	1.004 %
Standardization (Y/N)	-	0.605 %
User Perception (1-5 scale)	-	0.215%
Overall (1-5 scale)	-	0.423%
Latency	-	7.345 × slower
Time to First Token (TTFT) (s)	2.28±0.40	-

Table 2: Relative improvement of Multi-Agent Approach over Single Prompt Approach across quality and efficiency metrics.

ings and compare the approach with single prompt approach.

LLM-as-a-Judge: The LLM-as-a-Judge framework has been widely used in the literature to scale as well as explain the evaluation of LLM generated content (Chiang and Lee, 2023; Zheng et al., 2023; Mohammadi et al., 2025; Thakur et al., 2025). Chiang and Lee used LLM to evaluate the quality of texts in open-ended story generation and adversarial attacks tasks. They showed that LLM evaluation can produce results similar to expert human evaluation. Similarly, Thakur et al. showed that large models like GPT-4 Turbo, Llama-3.1;70B, and Llama-3;70B achieve stronger alignment with humans. Chiang et al. introduced Chatbot Arena for evaluating LLMs based on human preference. In this paper, we also employ LLM-as-a-Judge to evaluate generated job postings at scale, as it is more cost-effective than human evaluation while maintaining a high alignment with expert judgment.

7 Conclusions and Future Work

In this paper, we presented an LLM-based system for automated job posting generation that addresses the challenges of creating compliant, standardized, and engaging job descriptions at scale. We introduced a feedback-aware prompt refinement methodology that incorporates human-in-the-loop feedback to iteratively improve prompt quality. We compared two architectural approaches: single prompt and multi-agent, and evaluated their performance across quality metrics (compliance, standardization, user perception) and efficiency metrics (latency, time-to-first-token) using 1,056 generated job postings spanning nine job families and levels. Our evaluation demonstrates that while the multi-agent system achieves marginally higher quality scores (approximately 1% improvement), it incurs a substantial 7× latency penalty compared to the single-prompt approach. To bridge the gap between quality and real-time responsiveness, we implemented a streaming architecture for the single-prompt system, achieving a time-to-first-token well within the acceptable limits for real-world deployment.

As next steps, we plan to explore automated prompt refinement by learning from the edits made to the generated job postings by the hiring managers and recruiters before external publication. Specifically, we aim to automatically extract instructional patterns from these edits and incorporate them into the prompt optimization step. Additionally, rather than requiring explicit user input through forms, we plan to develop a conversational co-pilot interface where users can interact with the system through natural dialogue to iteratively refine job postings, enabling a more intuitive and efficient user experience.

Limitations

Our work has the following limitations. First, our current system generates job postings in a single pass, while professional writing typically involves iterative refinement. Enhancing the tool to support multi-turn interactions where users can iteratively refine generated content would better align with natural writing workflows. Second, translating collected feedback into prompt modifications requires human expertise to ensure compliance with organizational policies. Third, we evaluated only closed-source models (Claude Family and Nova Pro); the performance of open-source alternatives such as Llama (Touvron et al., 2023) or Mistral (Jiang et al., 2023, 2024) for job posting generation remains unexplored. Finally, our evaluation dataset is proprietary and cannot be publicly released due to confidentiality constraints. However, our methodology can be adapted to other organizational contexts.

Acknowledgments

We thank Janie Feinstein, Eric Ohrn, Saurabh Pant, Matt Knepper, Erica Ryan, Yue Wang, Jonathan Kristjansson, and Renchen Sun for their valuable discussions and insights throughout this work. We are grateful to the recruiting professionals and hiring managers who participated in our evaluation studies and provided critical feedback.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Amazon Artificial General Intelligence. 2024. The amazon nova family of models: Technical report and model card. *Amazon Technical Reports*.

Md. Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2024. MapCoder: Multi-agent code generation for competitive problem solving. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4912–4944, Bangkok, Thailand. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.

Bingxuan Li, Yiwei Wang, Jiuxiang Gu, Kai-Wei Chang, and Nanyun Peng. 2025. METAL: A multi-agent framework for chart generation with test-time scaling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30054–30069, Vienna, Austria. Association for Computational Linguistics.

Jiahui Li and Roman Klinger. 2025. iPrOp: Interactive prompt optimization for large language models with a human in the loop. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 276–285, Vienna, Austria. Association for Computational Linguistics.

Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. Prompt optimization with human feedback. *arXiv preprint arXiv:2405.17346*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. Evaluation and benchmarking of llm agents: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6129–6139, New York, NY, USA. Association for Computing Machinery.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing Wikipedia-like articles from scratch with large language models. 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 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement

learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. Many heads are better than one: Improved scientific idea generation by a LLM-based multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28201–28240, Vienna, Austria. Association for Computational Linguistics.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yangkun Wang, Zihan Wang, and Jingbo Shang. 2025. Direct prompt optimization with continuous representations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Vienna, Austria. Association for Computational Linguistics.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2025. Tradingagents: Multi-agents llm financial trading framework. *Preprint*, arXiv:2412.20138.

Cilin Yan, Jingyun Wang, Lin Zhang, Ruihui Zhao, Xiaopu Wu, Kai Xiong, Qingsong Liu, Guoliang Kang, and Yangyang Kang. 2025. Efficient and accurate prompt optimization: the benefit of memory in exemplar-guided reflection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–779, Vienna, Austria. Association for Computational Linguistics.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616.

Cheng Zhen, Ervine Zheng, Jilong Kuang, and Geoffrey Jay Tso. 2025. Enhancing LLM-as-a-judge through active-sampling-based prompt optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 960–970, Vienna, Austria. Association for Computational Linguistics.

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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.