STM JOURNALS

Journal of
Artificial Intelligence
Research & Advances

ISSN: 2395-6720
Volume 11, Issue 1, 2024
DOI Journal: 10.37591/JoAIRA

http://computers.stmjournals.com/index.php?journal=JoAIRA&page=index

Review JoAIRA

A Real-Time Visualization Framework to Enhance Prompt Accuracy and Result Outcomes Based on Number of Tokens

Prashant D. Sawant[1,*]

Abstract

In the rapidly evolving domain of artificial intelligence (AI), the efficacy of user- generated prompts has emerged as a critical factor influencing the quality of model- generated responses. Current methodologies for prompt evaluation predominantly rely on post-hoc analysis, which often leads to iterative prompting and increased computational overhead. Furthermore, the challenge of "prompt hallucinations," where AI models produce irrelevant or nonsensical responses, persists as a significant impediment to effective AI utilization. The present article introduces a novel framework that leverages real-time analysis of prompts (number of tokens to start with) to provide users with immediate feedback on prompt quality. The proposed system employs a dynamic scoring mechanism that assesses prompts against a comprehensive corpus and a set of predefined quality criteria, outputting a relative strength percentage or a 1-10 scale rating which can be displayed by colors (Red to Amber to Green). By integrating this framework into the AI interface, users can iteratively refine their prompts before submission, thereby enhancing the interaction efficiency and reducing the likelihood of hallucinatory outputs. This approach represents a paradigm shift from reactive to proactive prompt optimization, paving the way for more seamless and effective human-AI collaboration. Also, it may revolutionize the way users engage with AI systems, fostering a more productive and harmonious human-AI synergy.

Keywords: Real-time prompt analysis, prompt optimization, AI interaction efficiency, preventing prompt hallucination, human-AI collaboration

INTRODUCTION

The advent of large language models (LLMs) like GPT-3 and its successors has revolutionized the field of artificial intelligence, providing unprecedented capabilities in natural language processing. The effectiveness of these models is significantly influenced by the art of prompt engineering—the practice of crafting inputs that guide the AI to produce desired outputs. Prompt methods range from basic techniques like zero-shot and few-shot prompting to more advanced strategies such as chain-of-thought prompting, which have been instrumental in enhancing AI's problem-solving abilities [1–5].

*Author for Correspondence
Prashant D. Sawant
E-mail: prasdsaw@gmail.com

Director, Ai-D Consultancy, Melbourne, Australia

Received Date: March 26, 2024
Accepted Date: March 27, 2024
Published Date: April 05, 2024

Citation: Prashant D. Sawant. A Real-Time Visualization Framework to Enhance Prompt Accuracy and Result Outcomes Based on Number of Tokens. Journal of Artificial Intelligence Research & Advances. 2024; 11(1): 44–52p.

Despite these advancements, a phenomenon known as "AI hallucination" poses a challenge to the reliability of LLMs [6]. AI hallucinations occur when a model generates outputs that are nonsensical or unrelated to the prompt, often due to insufficient or biased training data, overfitting, or the use of complex, ambiguous prompts [7–11]. These inaccuracies can have serious implications, especially when AI is used in critical domains such as healthcare or legal services. To combat hallucinations, researchers have developed various

remedies, including Retrieval Augmented Generation (RAG) and other frameworks that integrate external data or employ structured prompting techniques to enhance the model's accuracy and robustness [12–14]. These methods aim to provide the AI with a richer context and reduce its reliance on potentially flawed pre-trained knowledge.

Google [15] has suggested best practices of prompting as follows.

- Clearly communicate what content or information is most important.
- Structure the prompt: Start by defining its role, give context/input data, then provide the instruction.
- Use specific, varied examples to help the model narrow its focus and generate moreaccurate results.
- Use constraints to limit the scope of the model's output. This can help avoid meanderingaway from the instructions into factual inaccuracies.
- Break down complex tasks into a sequence of simpler prompts.
- Instruct the model to evaluate or check its own responses before producing them.

The current landscape of AI prompting methods showcases a blend of successes and drawbacks. On one hand, LLMs have achieved remarkable feats in generating creative content,summarizing complex information, and even coding. On the other hand, issues like factual inaccuracies, ethical concerns, and the cognitive load on users to craft effective prompts remainsignificant bottlenecks [16,17].

However, human diversity and the multifaceted nature of human characteristics, including a multitude of languages and viewpoints, can give rise to a spectrum of errors in prompt creationand may preclude the formulation of prompts that garner universal acceptance as shown in Figure 1. Some of the intrinsic and extrinsic human factors that contribute to the spectrum of errors in prompt creationare stemming from:

- *Cultural Nuances:* Variations in cultural context can lead to misinterpretations orinappropriate content.
- *Socioeconomic Factors:* Disparities in wealth and education can affect the understandingand construction of prompts.
- *Regional Idioms:* Localized expressions may not translate well across different regions,affecting clarity.
- *Personal Biases:* Individual prejudices can inadvertently influence the tone and directionof prompts.
- *Historical Context:* Lack of awareness of historical sensitivities can result in prompts thatare offensive or inaccurate.
- *Psychological Divergence:* Differences in mental models and cognitive biases can skewthe intent of prompts.
- *Ethical Standards:* Divergent moral values can lead to conflicting views on whatconstitutes an acceptable prompt.
- *Linguistic Limitations:* Language proficiency levels can impact the ability to craftcoherent and precise prompts.
- *Technological Access:* Varying degrees of access to technology can lead to unequal opportunities for prompt optimization.
- *Physical Abilities:* Disabilities may affect the interaction with AI systems and the formulation of prompts.

These parameters [18-21] highlight the complexity of creating prompts that are both effective and sensitive to the vast array of human differences. They underscore the need for inclusive designand adaptive AI systems that can accommodate and understand this diversity.

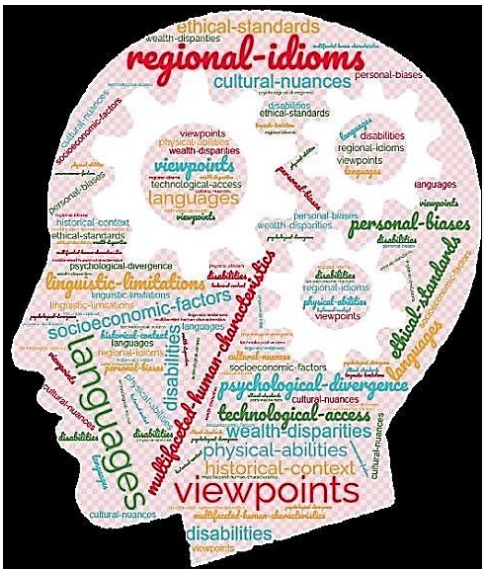

Figure 1. Intrinsic and extrinsic human factors that contribute to the spectrum of errors inprompt creation.

In light of these challenges, the new method proposed in this article offers a promising solution.It suggests a real-time prompt evaluation system that provides users with immediate feedbackon the quality of their prompts, expressed as a percentage or a scale rating. This innovative approach aims to streamline the prompting process, reduce the cognitive burden on users, andminimize the occurrence of hallucinations by ensuring that prompts are clear, specific, and well-structured before they are submitted to the AI.

BEST PRACTICES, FRAMEWORKS, AND QUALITY TESTING

Factors to Establish Best Practices

To establish the best practices for prompt design in chatbots, it is important to consider severalkey factors to ensure the prompts are effective and elicit the desired responses. Some of the best practices are as follows.

- *Clarity:* Prompts should be clear and unambiguous to guide the chatbot's responseaccurately.
- *Relevance:* Ensure prompts are relevant to the chatbot's capabilities and the user's intent.
- *Brevity:* Keep prompts concise to maintain user engagement and prevent overwhelmingthe chatbot.
- *Specificity:* Specific prompts lead to more precise and useful responses.
- *Contextual Awareness:* Include necessary context to make the conversation flow naturally.
- *User Intent:* Clearly convey the action or information desired from the chatbot.
- *Testing and Iteration:* Regularly test prompts with various inputs and refine prompts basedon outcomes.
- *Feedback Mechanism:* Incorporate user feedback to continuously improve prompt design.
- *Avoid Jargon:* Use simple language that all users can understand.
- *Conversational Tone:* Match the tone with the chatbot's persona for a human-likeinteraction.
- *Prompt Variability:* Use diverse types of prompts to keep the conversation dynamic.

By adhering to these best practices, we can design effective prompts that can enhance the performance of the chatbots and provide a better experience for users.

Testing the Quality of Prompts

Testing the quality of prompts in the chatbot is crucial for ensuring that it understands andresponds to user inputs effectively.

Some steps to test the quality of chatbot's prompts are as follows.

- *Define Objectives:* Clearly define the intended results that we need achieve with each prompt, determine the expected outcomes and how they align with the chatbot's goals.
- *Create Test Cases:* Develop a set of test cases that cover a wide range of scenarios, including common queries, edge cases, and potential misunderstandings.
- *Use Tools:* Using tools like promptfoo, which can help systematically test prompts, models, and RAGs answers with predefined test cases [22].
- *Manual Testing:* Conduct manual testing by entering prompts into the chatbot and evaluating the responses for accuracy, relevance, and helpfulness.
- *User Testing:* Perform user acceptance testing or UAT with real users to gather feedbackon the chatbot's performance in real-world scenarios [23].
- *Analyze Responses:* Review the chatbot's responses to different prompts and check if they meet the defined objectives. Look for patterns in any incorrect or suboptimal responses.
- *Iterate and Improve:* Use the insights gained from testing to refine the prompts and adjust them based on the chatbot's performance and user feedback.
- *Monitor Continuously:* Regularly analyze chatbot logs and user interactions to identifyareas for improvement and ensure the chatbot remains effective over time [24].

By following these steps, we can systematically evaluate and enhance the quality of chatbot's prompts, leading to a better user experience.

Defining Quality Metrics for AI Prompts

Defining quality metrics for AI prompts is a crucial step in ensuring that the AI model generatesrelevant and accurate responses. The following are some of the steps that can be considered toestablish the quality metrics.

- *Relevance:* The prompt should align with the intended purpose and context, and it shouldguide the AI to produce outputs that are topically relevant to the query.
- *Coherence:* The AI's response should be logical, well-structured, and easy to understand,and it should follow a clear and consistent train of thought.
- *Accuracy:* The information provided in the AI's response should be correct and up todate for prompts that require factual information.
- *Bias:* The AI's response should be free from unintended biases or stereotypes ensuringthe language and content are neutral and inclusive.
- *Efficiency:* Evaluate the average time or length required to generate an output as a goodprompt should lead to efficient generation of responses without unnecessary verbosity.
- *Groundedness:* The responses should be grounded in reality, meaning they should bebased on evidence and facts rather than speculation or fiction.
- *Fluency:* The output should be fluent, with proper grammar and syntax, making itreadable and understandable.
- *Similarity:* For tasks like summarization or translation, the output should closely matchthe expected result or ground truth in terms of content and meaning.
- *Objectivity:* The output should maintain a neutral tone and avoid subjective or biasedwords unless the prompt specifically requires an opinion.

Automation Tools to Assess Prompts

Thus, automated tools can quickly assess prompts based on objective criteria like coherence, fluency, and relevance, while human evaluators can provide insights into more subjective aspects like bias and groundedness. The combination of these methods can provide a comprehensive assessment of prompt quality. In this regard, platforms like Azure AI Studio [24] offer some features for monitoring and evaluating prompt quality, including the application ofResponsible AI evaluation metrics [25-27].

Azure AI Studio offers features like Prompt Flow, which can be used to build, benchmark, evaluate, and deploy real-time inference endpoints. Additionally, open-source tools like promptfoo provide systematically test prompts and evaluate their quality [22]. While promptfoo is not a real-time system, it can provide insights intohow we might structure our own system.

AutoPrompt [28] is a framework that enhances prompts for real-world use cases by automatically generating high-quality prompts tailored to user intentions. It uses a calibration process to iteratively build a dataset of challenging edge cases and optimizes the prompt accordingly. TheAutoPrompt method can be useful for masked language models without the need for additionalparameters or fine-tuning and has shown promising results in tasks such as sentiment analysisand natural language inference.

Above mentioned tools can significantly reduce manual effort in prompt engineering and effectively address common issues such as prompt sensitivity and inherent prompt ambiguity.These tools are a part of a growing ecosystem of resources aimed at empowering users to produce high-quality robust prompts using the power of large language models.

However, an important drawback of the above-mentioned frameworks and tools are that they do not provide real-time visualization of the prompt quality based on number of tokens for end-users who may not be experienced well in prompt engineering.

The Proposed Framework

The concept of displaying prompt quality as a percentage or on a scale during the writing process is an interesting idea. It would involve real-time analysis of the prompt against a set ofcriteria or a corpus to determine its effectiveness in eliciting the desired response from a language model like GPT (e.g., Copilot or ChatGPT). Currently, there is not a standard featurewithin GPT models that provides this functionality. However, theoretically, it could beimplemented by developing a custom tool or plugin that:

- *Analyzes the Prompt:* The tool would need to analyze the prompt in real-time as it isbeing written.
- *Compares to a Corpus:* It would compare the prompt to a large corpus of effectiveprompts and their outcomes to determine the quality.
- *Calculates a Score:* Based on this comparison, the tool would calculate a score thatreflects the prompt's relative strength or quality.
- *Displays the Score:* The score would then be displayed to the user, helping them adjustthe prompt to improve its quality before submission.

To create such a tool, we would need to consider the following steps.

- Define what constitutes a 'quality' prompt.
- Develop an algorithm to evaluate prompt quality.
- Train the tool on a dataset of prompts and their effectiveness.
- Integrate this tool into the user interface where prompts are written.

This tool would require access to a substantial corpus of data and a well-defined set of metricsfor evaluating prompt quality. It would also need to be fast enough to provide feedback in real-time without disrupting the user's workflow. Additionally, having a system that evaluates the quality of prompts in real-time and provides feedback could indeed help reduce instances of "prompt hallucinations," where an AI model generates outputs that are nonsensical or unrelatedto the prompt. By ensuring that prompts are clear, specific, and well-structured, users can guidethe AI to produce more accurate and relevant responses. While this is not a feature currently available, it is a valuable suggestion for future development in the field of AI and natural language processing. It could indeed streamline the process of interacting with AI models andenhance the user experience by reducing the need for iterative prompting.

RESULTS AND DISCUSSION

Implementing a Real-time Prompt Quality Display System

Implementing a real-time prompt quality display system into our AI model involves several steps, which include integrating a feedback mechanism that assesses the quality of prompts asthey are being written.

A high-level overview of this approach is as follows.

- *Define Quality Metrics:* Determine what constitutes a 'good' prompt for the specific usecase based on clarity, specificity, likelihood of generating accurate responses, etc.
- *Develop an Evaluation Algorithm:* Create an algorithm that can score prompts in real-time according to the defined metrics and it could involve natural language processing techniques and machine learning models trained on a dataset of effective prompts and their outcomes.
- *Integrate with AI Model:* The evaluation system should be integrated into the environment where prompts are entered, providing immediate feedback to the user. Thiscould be done through a plugin or an API that interacts with the AI model.
- *User Interface Design:* Design a user interface that displays the prompt quality score ina clear and non-intrusive manner, such as a percentage or a 1-10 scale next to the promptinput field.
- *Test and Iterate:* Before full deployment, test the system with actual users to gather feedback and refine the evaluation algorithm and user interface.
- *Deploy and Monitor:* Once the system is fine-tuned, deploy it for all users and continuously monitor its performance, adjusting as needed based on user feedback and changes in model behavior.

The key to a successful implementation is a well-defined set of quality metrics and a user- friendly interface that provides meaningful and actionable feedback without overwhelming theuser.

Example of Real-time Prompt Quality Monitoring System

A simple HTML code created that includes a script to change the color of the top bar based onthe text strength when the text type in the chatbot. The colors will transition from red to amberto green as the number of characters increases.

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Simple Chatbot</title>
<style>
   body { font-family: Arial, sans-serif; }
   #chatbot { width: 300px; margin: auto; }
   #top-bar { height: 20px; }
   #chat-input { width: 100%; }
</style>
</head>
<body>
<div id="chatbot">
   <div id="top-bar" style="background-color: red;"></div>
   <input type="text" id="chat-input" placeholder="Type your message here..." oninput="changeTopBarColor(this.value)">
   <!-- Add your chat messages here -->
</div>

<script>
function changeTopBarColor(text) {
   var topBar = document.getElementById('top-bar');
   // Define the color 'amber' as it's not a standard CSS color
   var amber = '#FFBF00';

   if (text.length < 10) {
     topBar.style.backgroundColor = 'red';
   } else if (text.length >= 10 && text.length < 20) {
     topBar.style.backgroundColor = amber;
   } else {
     topBar.style.backgroundColor = 'green';
   }
}
</script>
</body>
</html>
```

Figure 2. HTML code to display color code as a function of tokens.

The code (Figure 2) is as follows

It generated an HTML file that runs on Google Chrome for the demonstration purpose.The results are displayed below.

No text written in the chatbot which displayed red bar on the top. This is the initial stateof the chatbot when an inexperience user starts using the chatbot as shown in Figure 3.

Figure 3. Red color bar display for zero tokens.

A text with three words (tokens) written which are not sufficient for the prompting as shown in Figure 4.

Figure 4. Red color bar display for three tokens.

Thus, the inexperience user will know that that their prompt is insufficient to get meaningful results.

When the words increase further to four words the display color changes to amber which will inform the inexperienced user that the prompt has improved but notsufficient to get the decent results as shown in Figure 5.

Figure 5. Amber color bar display for four tokens

When the words increase further to 5-7 words the display color changes to green colorindicating that the prompt is of minimum superior quality to get a decent quality result as shown in Figure 6.

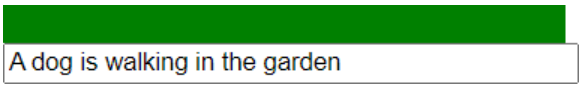

Figure 6. Green color bar display for more than five tokens.

This is a low-profile example to demonstrate the principle that can be embedded in thepopular chatbots like copilot or ChatGPT.

Benefits of a real-time prompt quality display system

- *Reduced Cognitive Load:* Users can focus on their objectives rather than the mechanicsof prompt crafting.
- *Enhanced Learning:* Real-time feedback facilitates a learning loop for users, helpingthem understand how different prompts influence AI behavior.
- *Increased Efficiency:* Immediate feedback can reduce the time spent on trial and error,leading to quicker and more productive AI interactions.
- *Improved Model Understanding:* Users gain insights into how the model processes information, which can inform better prompt design.
- *Quality Control:* Helps maintain a consistent level of quality in prompts, which is particularly important in professional or commercial settings.

CONCLUSIONS

The present article demonstrates a simple color display framework that can be added to chatbots like Copilot or ChatGPT that can help inexperienced or experienced users to understand the quality of their tokens to some extent. This framework can be further modifiedto include various other parameters that are related to the carpus searches. Such a system couldsignificantly enhance the user experience and the overall utility of AI models in various applications.

REFERENCES

1. X. Amatriain, Prompt Design and Engineering: Introduction and Advanced Methods. *arXiv*:2401.14423 (2024).
2. L. Huang, W. J. Yu, W. T. Ma, W. H. Zhong, Z. G. Feng, H. T. Wang, Q. G. Chen, W. H. Peng, X. C. Feng, B. Qin, and T. Liu, A Survey on Hallucination in Large LanguageModels: Principles, Taxonomy, Challenges, and Open Questions. *arXiv*:2311.05232 (2023).
3. H. Sun, An RL Perspective on RLHF, Prompting, and Beyond. *arXiv*:2310.06147 (2023).
4. M. Mosbach, Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation. *arXiv*:2305.16938 (2023).
5. S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, Chain-of-Verification Reduces Hallucination in Large Language Models. *arXiv*:2309.11495 (2023).
6. M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, and T. Hoefler. Graph of Thoughts:Solving Elaborate Problems with Large Language Models. *arXiv*:2308.09687 (2023).
7. What are AI Hallucinations? IBM, https://www.ibm.com/topics/ai-hallucinations [Access on 14 March 2024].
8. M. Salvagno, F. S. Taccone and A. G. Gerli, Artificial intelligence hallucinations. *Crit.Care* 27, 180 (2023).
9. S. V. Bentley and C. Naughtin, Both Humans and AI Hallucinate - But Not in the SameWay. (2023). *https://www.csiro.au/en/news/All/Articles/2023/June/humans-and-ai- hallucinate* [Access on 14 March 2024]
10. N. Maleki, B. Padmanabhan and K. Dutta, AI Hallucinations: A Misnomer Worth Clarifying. *arXiv*:2401.06796 (2024).
11. A. Bruno, P. L. Mazzeo, A. Chetouani, M. Tliba, and M. A. Kerkouri, Insights into Classifying and Mitigating LLMs' Hallucinations. *arXiv*:2311.08117 (2023).
12. S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. *Trans. Associat. for Computational Linguistics* 2023; 1,1 1 - 17.
13. Y. F. Gao, Y. Xiong, X. Y. Gao, K. X. Jia, J. L. Pan, Y. X. Bi, Y. Dai, J. W Sun, Q. Y.Guo, M. Wang, and H. Wang, Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv*:2312.10997.
14. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv*:2005.11401.
15. Prompt Engineering for Generative AI. *https://developers.google.com/machine-learning/resources/prompt-eng*. [Access on 14 Mar 2023].
16. A. Skulmowski and K. M. Xu, Understanding Cognitive Load in Digital and Online Learning: A New Perspective on Extraneous Cognitive Load. *Educ. Psychol. Rev.* 34, 171 - 196 (2022).
17. A. Trotta, M. Ziosi and V. Lomonaco, The Future of Ethics in AI: Challenges and Opportunities. *AI & Soc* 38, 439 - 441 (2023).
18. R. A. Shams, D. Zowghi and M. Bano, AI and the Quest for Diversity and Inclusion:A Systematic Literature Review. *AI and Ethics* (2023).
19. R. Crowell, Why AI's Diversity Crisis Matters, and How to Tackle It. *Nature* (2023).
20. A. Howard and C. Isbell, Diversity in AI: The Invisible Men and Women. *MIT Sloan Management Review* (2020).

21. D. Zowghi and F. da Rimini, Diversity and Inclusion in Artificial Intelligence. *arXiv:2305.12728*, (2023).
22. S. Collins, How to Use Promptfoo for LLM Testing. *The Deep Hub* Feb, 2024, Medium; https://medium.com/thedeephub/how-to-use-promptfoo-for-llm-testing-13e96a9a9773 [Access on 16 March 2024].
23. H. K. N. Leung and P. W. L. Wong, A study of User Acceptance Tests. *Software QualityJ*. 6, 137 - 149, (1997).
24. V. Alto, Evaluating LLM-Powered Applications with Azure AI Studio. *Medium* https://medium.com/microsoftazure/evaluating-llm-powered-applications-with-azure- ai-studio-b3cec3eba322 [Access on 16 March 2024].
25. B. Xia, Q. Lu, L. Zhu , S.U. Lee, Y. Liu, and Z. Xing, Towards a Responsible AI Metrics Catalogue: A Collection of Metrics for AI Accountability. *Semantic Scholar*, Corpus ID: 265352192, (2023).
26. G. Berman, N. Goyal and M. Madaio, A Scoping Study of Evaluation Practices for Responsible AI Tools: Steps Towards Effectiveness Evaluations. *arXiv*:2401.17486 (2024).
27. B. M. Xia, Q. H. Lu, L. M. Zhu, S. U. Lee, Y. Liu and Z. C. Xing, Towards a Responsible AI Metrics Catalogue: A Collection of Metrics for AI Accountability. *arXiv*:2311.13158, (2023).
28. T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace and S. Singh, AutoPrompt: ElicitingKnowledge from Language Models with Automatically Generated Prompts. *arXiv*:2010.159801, (2020).

