

Tools in the Loop: Quantifying Uncertainty of LLM Question Answering Systems That Use Tools

Extended Abstract

Panagiotis Lymperopoulos
Tufts University
Medford, United States
plympe01@tufts.edu

Vasanth Sarathy
Tufts University
Medford, United States
vasanth.sarathy@tufts.edu

ABSTRACT

Modern Large Language Models (LLMs) increasingly rely on external tools—such as classifiers and knowledge retrieval systems—to deliver accurate answers when their pre-trained knowledge falls short. While this integration broadens their utility, it also raises a critical issue: ensuring the trustworthiness of the combined outputs. In high-stakes settings like medical decision-making, it is vital to evaluate uncertainty in both the LLM’s response and the external tool’s output. In this work we introduce a novel framework that jointly assesses the combined uncertainty of the LLM and its external tools and derive practical and effective approximations to estimate uncertainty. Our approach is validated on two synthetic QA datasets and an experiment with retrieval-augmented generation (RAG) systems, demonstrating enhanced reliability when external information is required for the LLM to produce answers.

KEYWORDS

LLM, tools, uncertainty, generative AI, agents

ACM Reference Format:

Panagiotis Lymperopoulos and Vasanth Sarathy. 2025. Tools in the Loop: Quantifying Uncertainty of LLM Question Answering Systems That Use Tools: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, Detroit, Michigan, USA, May 19 – 23, 2025, IFAAMAS, 3 pages.

1 INTRODUCTION

Large Language Models (LLMs) are increasingly integrated with external tools to enhance reasoning and access information beyond their training data [11], extending their applicability to critical domains such as medicine [14] and law [5]. However, existing uncertainty quantification (UQ) methods [4] focus only on the LLM’s outputs and fail to account for the uncertainty introduced by tool calls. This limitation is particularly problematic in specialized domains where external tools provide essential information.

We propose a framework for quantifying uncertainty in tool-calling question-answering (QA) systems by jointly modeling the uncertainty of both the LLM and the external tool. Assuming a white-box setting where tool uncertainty is known, we extend semantic entropy [4] to this scenario and introduce an efficient approximation for practical deployment. We evaluate our method

on two synthetic QA datasets derived from the IRIS [3] and PIMA diabetes datasets, as well as on a retrieval-augmented generation (RAG) [6] task using the BoolQ dataset [1]. Our results demonstrate the effectiveness of our framework in improving uncertainty estimation in tool-augmented LLM systems.

2 RELATED WORKS

Uncertainty quantification for LLMs has been explored through supervised models trained on LLM logits [7] and semantic uncertainty approaches [2, 4], which estimate uncertainty over meanings rather than individual tokens. While these techniques are effective in detecting hallucinations, they do not consider the additional uncertainty introduced by tool calls. Our framework extends these methods by incorporating the predictive uncertainty of external tools into a unified model.

Research on tool-calling LLMs [10, 11] has primarily focused on training models for structured tool use and developing datasets to enhance tool-selection capabilities [8]. Tool-calling benchmarks [9, 15] evaluate models on API calls, database queries, and retrieval tasks but do not address uncertainty quantification. Additionally, many of these benchmarks use deterministic or highly complex tools, making uncertainty estimation difficult. Instead, we focus on controlled QA tasks with tools that provide known uncertainty estimates, enabling a principled study of UQ in tool-augmented LLMs.

3 METHOD

In this section we present our framework for uncertainty quantification in tool-using LLMs. Let \mathcal{S} be the set of all token sequences. Let $x \in \mathcal{S}$ be the prompt to the LLM. Let $a \in \mathcal{A} \subset \mathcal{S}$ be a sequence representing an invocation of an external tool. Let $z \in \mathcal{Z}$ be a result produced by the external tool, where \mathcal{Z} is the space of possible responses (e.g. $\mathcal{Z} = \{0,1\}$ for binary classifiers). Finally, let $y \in \mathcal{S}$ be a sequence corresponding to the response produced by the LLM after receiving the prompt and invoking the tool. Note that this formulation covers the case of multiple tools, as they can be encapsulated into a single tool with a more complex tool call a .

Our framework makes the following assumptions: 1) The final response y is independent of the invocation of the tool a given the tool response z . 2) The predictive entropy $H(z|a)$ of the tool $p(z|a)$ is known.

We model the tool-calling process as a sequential process encompassing two calls to the LLM and one call to the tool.

$$p_{\theta}(y, z, a|x) = p_{\theta}(y|z, x)p(z|a)p_{\theta}(a|x), \quad (1)$$



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

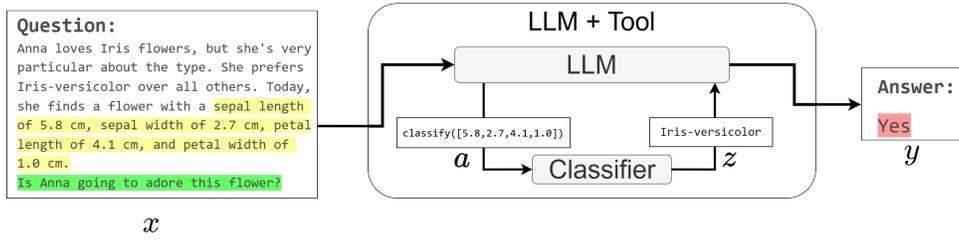


Figure 1: Illustration of an LLM+tool system. The system receives an input prompt x that requires a tool (e.g. a classifier) to answer. The LLM produces a tool call a which acts as input to the tool, which produces output z . Finally, the LLM produces the final answer y . Yellow indicates the features used to call the tool, green the question and red the final answer of the combined system.

Model	STA _S	STA _P	SEFA	PEFA	Model	STA _S	STA _P	SEFA	PEFA	Model	STA _S	STA _P	SEFA	PEFA
Llama-3-8B-Inst.	0.845	0.824	0.615	0.553	Llama-3-8B-Inst.	0.791	0.730	0.663	0.634	Llama-3.1-8B-Inst.	0.675	0.646	0.662	0.622
Llama-3.1-8B-Inst.	0.752	0.668	0.642	0.529	Llama-3.1-8B-Inst.	0.675	0.662	0.515	0.580	Llama-3-8B-Inst.	0.648	0.645	0.570	0.575
Mistral-7B-Inst.	0.786	0.718	0.667	0.605	Mistral-7B-Inst.	0.782	0.702	0.516	0.570	Mistral-7B-Inst.	0.668	0.705	0.502	0.711

Table 1: Combined results for IRIS (left), Diabetes (middle), RAG (right) showing AUROC of STA_S, STA_P, SEFA, PEFA. Higher numbers indicate better correlation with correctness.

Equation (1) shows the joint distribution over the variables in the system, where θ corresponds to the parameters of the LLM. Figure 1 illustrates our framework for modeling tool-calling LLM systems.

3.1 Uncertainty quantification for tool-calling systems.

Within our framework, we quantify uncertainty using entropy. We now present a derivation for the predictive entropy $H(y|x)$ in our framework in terms of the known $H(z|a)$ and other terms.

$$\begin{aligned}
 H(y|x) &= H(y, z, a|x) - H(z, a|x, y), \\
 H(y|x) &= H(y|z, a, x) + H(z|a, x) + H(a|x) \\
 &\quad - H(z|x, y, a) - H(a|x, y).
 \end{aligned}$$

By the conditional independence in eq. (1) we obtain:

$$H(y|x) = H(y|z, x) + H(z|a) + H(a|x) - H(z|y, a) - H(a|x, y). \quad (2)$$

Similarly, we can also derive semantic entropy [4]:

$$H(C|x) = H(C|z, x) + H(z|a) + H(a|x) - H(z|y, a) - H(a|x, y), \quad (3)$$

where $H(C|z, x)$ can be estimated with samples. Computing equations (2),(3) is not tractable due to the posterior distributions. However, we can still obtain efficient and useful uncertainty measures under some additional assumptions: **(a) The answer y depends strongly on the tool output z** and z is easy to infer given y ($H(z|y, a) \approx 0$). **(b) All of the information a needed for the tool is contained in x** , so knowledge of y does not help in inferring a ($H(a|x) - H(a|x, y) \approx 0$).

These additional assumptions allow us to simplify equations (2) and (3) into the Strong Tool Approximation (STA) of Predictive and Semantic Entropy (STA_P, STA_S):

$$STA_P(x) = H(y|z, x) + H(z|a), \quad (4)$$

$$STA_S(x) = H(C|z, x) + H(z|a). \quad (5)$$

These metrics are simple to compute, amounting to only additively combining the entropy of the LLM’s final answer, which can be estimated using existing methods, and the entropy of the tool response, which is assumed to be known. In the case of typical machine learning tools such as classifiers or regression models, this can be directly computed as the entropy of the output distribution. Notably, these metrics also apply to RAG systems by treating the retriever as a categorical distribution over documents and computing the entropy of the distribution.

4 RESULTS

In this section we validate our framework and the derived metrics. In our experiments we use three synthetic QA datasets that require tools: IRIS QA, PIMA QA and a small RAG dataset. The first two are derived from well known machine learning datasets [3, 12] and pose the corresponding classification problem as a natural language question (fig. 1). The RAG dataset consists of yes/no questions from the BoolQA [1] dataset, with a document bank derived from wikipedia [13]. We evaluate on 150 questions from each dataset.

We estimate uncertainty using our STA metrics and compare against baseline semantic and predictive entropies over the LLM final answer (*SEFA, PEFA*). We evaluate on 3 pre-trained instruction tuned models: Meta’s llama 3.0 and 3.1 8B models and Mistral 7B. Table 1 summarizes the results, showing STA metrics outperform the baselines in almost every case. In the RAG experiment, performance gains are less clear because the STA assumptions are not fully met: making use of the retrieved information may require additional reasoning and knowledge in some cases. Still the metrics are more informative in most cases for little additional computation.

Acknowledgments. This research was supported in part by Other Transaction award HR00112490378 from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program.

REFERENCES

- [1] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *NAACL*.
- [2] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature* 630, 8017 (2024), 625–630.
- [3] R. A. Fisher. 1936. Iris. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C56C76>.
- [4] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664* (2023).
- [5] Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. 2023. Large language models in law: A survey. *arXiv preprint arXiv:2312.03718* (2023).
- [6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [7] Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty Estimation and Quantification for LLMs: A Simple Supervised Approach. *arXiv preprint arXiv:2404.15993* (2024).
- [8] Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, et al. 2024. ToolACE: Winning the Points of LLM Function Calling. *arXiv preprint arXiv:2409.00920* (2024).
- [9] Yun Peng, Shuqing Li, Wenwei Gu, Yichen Li, Wenxuan Wang, Cuiyun Gao, and Michael Lyu. 2021. Revisiting, Benchmarking and Exploring API Recommendation: How Far Are We? *arXiv:2112.12653* [cs.SE]
- [10] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Tool Learning with Large Language Models: A Survey. *arXiv preprint arXiv:2405.17935* (2024).
- [11] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 68539–68551. https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf
- [12] Jack W Smith, James E Everhart, WC Dickson, William C Knowler, and Robert Scott Johannes. 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*. American Medical Informatics Association, 261.
- [13] Noah A Smith, Michael Heilman, and Rebecca Hwa. 2008. Question generation as a competitive undergraduate course project. In *Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge*, Vol. 9.
- [14] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine* 29, 8 (2023), 1930–1940.
- [15] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems* 36 (2023), 50117–50143.