Evaluation of Function Calling Performance in an On-Premise LLM: A Comparison between AtomicGPT and Open LLMs

Joowon Cha[a,b] , Seungdon Yeom[a,b], Yonggyun Yu[a,b*]
[a]*Korea Atomic Energy Research Institute, 111, Daedeok-daero 989 beon-gil , Yuseong-gu , Daejeon, 34057, Korea*
[b]*Korea National University of Science & Technology, 217, Gajeong-ro, Yuseong-gu , Daejeon, 34113,Korea*
[*]*Corresponding author: ygyu@kaeri.re.kr*

Keywords : domain-specific language model, function calling, benchmark, nuclear, AtomicGPT

1. Introduction

The rapid advancement of large language models (LLMs) has significantly increased their adoption across diverse industries. Recently, research efforts have expanded beyond general-purpose models to develop domain-specific models tailored to particular sectors, including the nuclear industry. Especially in the security-critical nuclear domain, specialized models such as AtomicGPT[1,2] have emerged, designed for secure on-premise deployment. These specialized models have demonstrated their ability to enhance efficiency and reduce human errors in various tasks, such as report generation, regulatory review, and technical verification.

Moreover, AtomicGPT's potential extends beyond documentation tasks; it is expected to serve as an "AI reactor operator," capable of controlling nuclear plant simulators. In this context, the Function-Calling capability of LLMs is particularly critical, enabling direct interaction with complex computational and simulation tools. This capability ensures effective deployment in the nuclear industry, where safety and reliability are paramount.

This study compares and analyzes the Function-Calling performance of AtomicGPT against other open-source language models of similar parameters scale, utilizing metrics such as the ToolCorrectnessMetric and TaskCompletionMetric. These metrics were specifically designed to evaluate a model's accuracy and efficiency in complex function-calling scenarios. Ultimately, the goal of this study is to quantitatively validate AtomicGPT's function-calling performance and explore its potential as an optimized tool for specialized nuclear applications, such as future AI reactor operation.

2. Proposed Method

This Section introduces AtomicGPT-8B, a domain-specific model specialized for the nuclear industry, and discusses performance evaluation and comparative experiments using function-calling metrics

2.1 Overview of AtomicGPT

AtomicGPT-8B is a language model specialized for the nuclear domain developed by Korea Atomic Energy Research Institute(KAERI), designed to effectively acquire and utilize nuclear-related expertise. The model is based on Llama-3.1-8B and has been further pretrained and instruction-tuned using a comprehensive nuclear-domain-specific dataset.

The training dataset comprises high-quality texts collected from authoritative nuclear institutions and research organizations. Major data sources include Korea Hydro & Nuclear Power (KHNP)'s "Glossary of Nuclear Energy Terms[3]" and "Glossary of Nuclear Laws and Regulations[4]," the Nuclear Safety and Security Commission (NSSC)'s "Glossary of Nuclear Safety Regulations[5]," Korea Atomic Energy Research Institute (KAERI)'s "Academic Papers on Nuclear Energy" and "Recent Trends Announcements[6]", Seoul National University Nuclear Policy Center's "Nuclear Wiki[7]," and KAERI's "internal data," collectively forming approximately 41,000 training examples.

Leveraging this dataset, AtomicGPT-8B was optimized to generate sophisticated responses to a wide range of nuclear-related queries, including defining nuclear terminologies, interpreting relevant laws and regulations, and analyzing current technological trends. Additionally, the model demonstrates capabilities in summarizing expert-level technical documents and explaining complex nuclear concepts in an easily comprehensible manner.

In this study, we quantitatively evaluate the function-calling performance of AtomicGPT-8B and discuss its practical applicability within the nuclear industry and research communities.

2.2 Measuring and comparing performance across models using the Function Calling Metric

This section describes the experiments conducted to measure and compare the function calling performance of AtomicGPT-8B, a domain-specific language model for nuclear energy. First, we collected the data required for the evaluation to build a dataset that can be used for function calling evaluation. Then, we constructed a metric for measuring function calling performance and evaluated the models by measuring the difference between the expected output results and the actual output results. Finally, we compared the metric results of each model.

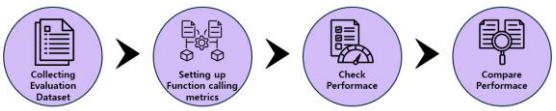

Fig 1. Performance measurement and comparison process across models

2.2.1 Collecting function calling evaluation data

To compare and analyse the function calling performance of AtomicGPT-8B, we first collected evaluation data to measure the performance. Referring to the IAEA's iPWR simulator documentation[8], the variable names were defined as functions, and the information of the variables was obtained from existing documents and web searches[9] to create a dataset suitable for the function calling dataset format.

The following is an example of the evaluation dataset, which shows the function schema to return the reactor output information:

Table 1: *Dataset for a function that returns reactor power output information*

Function Calling Evaluation Dataset Example

```
{
  "name": "get_reactor_power",
  "description": "Returns mock reactor power output
                  for a given reactor ID.",
  "parameters": {
    "type": "object",
    "properties": {
      "reactor_id": {
        "type": "string",
        "description": "The unique ID of the
                        reactor."
      }
    },
    "required": [
      "reactor_id"
    ]
  }
}
```

Using this function schema, various functions required for function calling performance evaluation were defined and composed into an evaluation dataset.

2.2.2 Function Calling Metric Configuration

To compare the function calling performance of AtomicGPT-8B, we selected metrics suitable for function calling performance evaluation and used them as evaluation metrics. The metrics selected for this purpose are two function calling metrics that are known to be useful for performance measurement: **ToolCorrectnessMetric**[10], which compares whether all tools expected to be used are actually called, and **TaskCompletionMetric**[11], which determines how well the assigned task is performed.

- **ToolCorrectnessMetric** is a metric that evaluates the correctness of the model's function/tool calling, calculated by comparing whether all the tools expected to be used were actually called. This comparison is done as follows Compute the cosine similarity of the function names, parameter keys, and parameter values of the expected and actual output results, respectively, using the 'all-MiniLM-L6-v2' model from the Python library sentence_transformer; compute the average of the cosine similarity for each value; and apply a weighted average-based normalisation. The resulting values are output as the final ToolCorrectness value.

- **TaskCompletionMetric** An alignment score to evaluate how well the model performed the assigned task, which is a metric to evaluate the completeness of the model in performing function/tool calling. This score is evaluated by composing three sub-alignments: Content Alignment, Performance Alignment, and Expression Alignment.
Content Alignment evaluates the semantic match between the user's query intent and the actual function call.
Performance Alignment considers whether the expected function was actually called and whether the parameters were passed correctly.
Expression Alignment evaluates whether the actual called function has the correct format and JSON structure. It is calculated by penalising for each error, starting with a base score of 1 and subtracting a certain amount for each incorrect format. The score cannot be negative, and the minimum value is zero.
A final score is calculated by weighted average of the three alignment scores calculated above.

2.2.3 Selection of Comparison Models and Performance Analysis

In this paper, we constructed an evaluation dataset to evaluate the function calling performance of AtomicGPT-8B and analysed the performance by comparing the evaluation metric results of selected open source models to compare the function calling performance of AtomicGPT-8B. The selected open source models are as follows:

- **meta-llama/Llama3.1-8B-Instruct**
- **mistralai/Mistral-7B-Instruct-v0.3**
- **Qwen/Qwen2.5-7B-Instruct**

In our performance analysis, we used two main metrics to evaluate tool call correctness, which is

important in function calling, and task completion: ToolCorrectnessMetric, TaskCompletionMetric.

2.2.4 Compare performance measures by model

As shown in the evaluation results graph, AtomicGPT-8B outperformed other models in ToolCorrectnessMetric with a score of about 0.63. In particular, Mistral-7B-Instruct-v0.3 had the lowest score of about 0.19, while Llama3.1-8b-Instruct and Qwen2.5-7B-Instruct had scores of about 0.44 and 0.29, respectively.

In TaskCompletionMetric, AtomicGPT-8B performed the best with a score of around 0.52, followed by Llama3.1-8b-Instruct with a score of around 0.48. Qwen2.5-7B-Instruct and Mistral-7B-Instruct-v0.3 performed similarly at around 0.38 and 0.36 respectively.

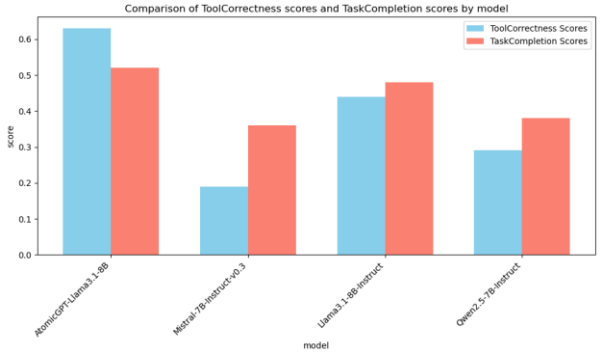

Fig 2. Comparsion of ToolCorrectness and TaskCompletion scores across open source models.

AtomicGPT-8B outperforms the other models (Llama3.1-8b-Instruct, Mistral-7B-Instruct-v0.3, and Qwen2.5-7B-Instruct) on all evaluation criteria, especially on the ToolCorrectness score, which shows that AtomicGPT-8B has an excellent ability to perform complex function calls accurately.

AtomicGPT-8B also scored well on TaskCompletionMetric, demonstrating its strong performance.

3. Conclusions

This study evaluates the function calling performance of AtomicGPT-8B, a domain model specialized for the nuclear energy domain, and compares it with several open source models. The evaluation results show that AtomicGPT-8B performs well on key performance metrics, demonstrating its potential as an effective tool for critical applications in the nuclear industry.

AtomicGPT-8B performed the best on both ToolCorrectnessMetric and TaskCompletionMetric among all the models evaluated, especially on ToolCorrectnessMetric, significantly outperforming the other open source models (Llama3.1-8b-Instruct, Mistral-7B-Instruct-v0.3, Qwen2.5-7B-Instruct).

This performance is attributed to the fact that AtomicGPT-8B is a domain-specific model that is further trained on a nuclear power dataset. Unlike general-purpose models, AtomicGPT-8B is optimized to produce sophisticated and reliable answers to specialized questions, such as interpreting nuclear terminology, interpreting regulations and laws, and analyzing technology trends.

The results of this study confirm that AtomicGPT-8B can be utilized in a variety of applications in the nuclear industry, ranging from document generation to acting as an “AI reactor operator” including simulator control. In particular, the Function Calling feature plays a key role in the nuclear field, where safety and reliability are critical, ensuring a high level of task accuracy and efficiency.

In conclusion, AtomicGPT-8B has shown promise as a nuclear domain-specific LLM and has the potential to become an optimized tool for the nuclear industry. With future research, it has the potential to be extended to advanced use cases such as real-time reactor simulation and automated compliance reviews, which will contribute to building a safer and more efficient nuclear operating environment.

ACKNOWLEDGMENTS

This work was supported in part by Korea Atomic Energy Research Institute R&D Program under Grant KAERI-524540-25.

REFERENCES

[1] Yeom Seung Don, ChangSu Choi, Lim KyungTae, & Yu Yong Gyun (2024-11-20). Development and Performance Evaluation of a Domain-Specific Language Model for Nuclear: A Comparative Study Using a Custom-Built Dataset. Proceedings of Symposium of the Korean Institute of communications and Information Sciences, Gyeongbuk.
[2] Yeom Seungdon, Kim Soyeon, & Yu Yonggyun (2024-10-24). Embedding-Based Response Blocking Algorithm for Enhancing the Reliability of Domain-Specific Language Models in the Atomic Energy Industry. *Transactions of the Korean Nuclear Society Autumn Meeting*, Changwon, Korea.
[3]한국수력원자력(주)_원자력용어집(2019). https://www.data.go.kr/data/15038485/fileData.do?recommendDataYn=Y (accessed Aug, 16, 2024)..
[4] 한국수력원자력(주)_원자력관련법령 용어집(2014). https://www.data.go.kr/data/15002295/fileData.do (accessed Aug, 16, 2024).
[5]원자력안전위원회 원자력안전규제용어사전. https://www.nssc.go.kr/ko/cms/FR_CON/index.do?MENU_ID=2460 (accessed Aug, 14, 2024).
[6] 한국원자력연구원_국내 원자력 관련 최신 동향 발표자료

목록(2020). https://www.data.go.kr/data/3077573/fileData.do (accessed Aug, 16, 2024).
[7]AtomicWiki (2023) https://atomic.snu.ac.kr/index.php/%EB%8C%80%EB%AC%B8 (accessed Aug, 16, 2024).
[8] Vienna (2017) https://www-pub.iaea.org/MTCD/Publications/PDF/TCS-65_web.pdf
[9] IAEA | PRIS Power Reator information System https://pris.iaea.org/pris/home.aspx (updated Mar 16. 2025)
[10] Jeffrey *Ip* *DeepEval-ToolCorrectness*Metric https://docs.confident-ai.com/docs/metrics-tool-correctness (updated Mar 16, 2025)
[11] Jeffrey *Ip* *DeepEval-TaskCompletionMetric* https://docs.confident-ai.com/docs/metrics-task-completion (updated Mar 16, 2025)