# Guidelines for Quality Assurance of AI-based Products and Services

Excerpt of the QA4AI Guidelines :
Chapter on LLM and Conversational Generative AI

**QA4AI Consortium**

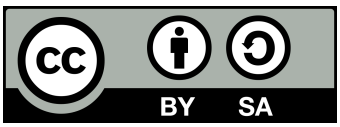
Consortium of Quality Assurance for
Artificial-Intelligence-based Products and Services

LLM and Conversational Generative AI Working Group
QA4AI Consortium
Japan
2024-03

## About

These guidelines are issued by the AI Product Quality Assurance Consortium (QA4AI Consortium) for the quality assurance of conversational generative AI products.
Given the developmental state of conversational generative AI, its technologies, and applications, these guidelines are not intended to be exhaustive or complete. Therefore, these guidelines serve as a reference for reflecting upon one's domain, company, or organizational circumstances.
The AI Product Quality Assurance Consortium, or any individuals or entities belonging to the consortium, do not have any responsibilities for the quality of AI products developed or offered in accordance with these guidelines. Furthermore, these guidelines does not represent the official view of any individuals or entities belonging to the consortium or their respective organizations or any higher-level entities.





# Contents

# 1. Introduction

Artificial-Intelligence (AI) products are being developed and used by a variety of organizations, and potential risks and quality issues exist among them. There was strong demand in the industry for guidelines on how risks should be managed and quality should be ensured. Given this context, we founded a consortium named as Quality Assurance for Artificial-Intelligence-based products and services (QA4AI) in 2018 to address this challenge. We have published the AI Product Quality Assurance Guidelines (QA4AI Guideline) and its updates since 2019.

The QA4AI Guideline is a set of guidelines that outline the risks and quality assurance techniques for AI products. The features of the guideline include five axes for AI product quality (Data Integrity, Model Robustness, System Quality, Process Agility, and Customer Expectation), a technical catalogue, and domain-specific sub-guideilnes for concrete domains of "Content Generation Systems," "Voice User Interface," "Industrial Processes," "Automated Driving," "AI-OCR," and "Large-language models and conversational generative AI."

Since the release of ChatGPT in 2022, a lot of AI products and models have been widely and rapidly released on the basis of large language models (LLM) and conversational generative AI. However, the quality and trust aspects of generative AI have only just begun to be discussed. Our QA4AI consortium has responded to this issue and extended its guidelines to deal with quality assurance of large language models and conversational generative AI.

The following chapter (Chapter 2) is an excerpt of QA4AI Guidelines for large language models and Conversational Generative AI, from Version 2024.01.

# 2. Large Language Models and Conversational Generative AI

This chapter focuses on quality assurance for generative AI, especially for systems using large language models (LLM). While traditional AI systems are trained for a specific task that follows a fixed set of inputs and outputs as supervised learning, LLM-based generative AI systems are trained for a general set of sentences and dialogues. The result is a general-purpose conversational generative AI that can handle a wide variety of tasks in response to instructions in inputs called prompts. Although the potential to handle a wide variety of use cases with relatively little effort is being pursued, the characteristics of the implementation of such systems have led to quality issues, including responses that are natural but do not match the instructions or facts (hallucination). This chapter provides guidelines on quality assurance for LLM-based, conversational generative AI systems.

At the time of writing this guideline (late 2023), we are focusing on a type of generative AI system of text input/output. We try to provide a general description, but concrete examples and techniques deal primarily with textual input/output. Multimodal systems that handle diagrams and images will be discussed in future versions.

## 2.1 Overview of LLM and Conversational Generative AI

### 2.1.1 configuration and operation

The terminology used in this chapter is defined in Table 2.1. We describe the prerequisite knowledge about LLM, which is the foundation of conversational generative AI.

Table 2.1: Definitions of Terminology

| Terminology | Meaning and Explanation |
|---|---|
| Generative AI | Generative AI is a generic term for AI technologies that generate data, such as text and images, by learning the distribution of target data. Conversational generative AI is based on LLM and allows for control of output via inputs called prompts. This chapter discusses such AI. |
| LLM: Large Language Model | A language model that treats the probability of occurrence of sentences and words as a deep-learning model and constructed using very large amounts of training data. Examples of LLM include OpenAI's GPT, Google's PaLM, and Meta's LLaMA. |
| Foundational Model | A highly versatile model obtained through large training using general training data that is independent of specific tasks, called pre-training. It can be used to construct individual models for specific downstream tasks through fine tuning and other customization. |
| Downstream Task | For a pre-trained foundation model with a high degree of generality, additional learning may be conducted for it to adapt to more specific tasks such as translation, summarization, classification, and logical reasoning. In this chapter, we refer to them simply as tasks. |

| | |
|---|---|
| Fine-tuning | The process of training a highly generalized pre-trained model for a particular downstream task. It is often possible to construct a high-performance model for a specific task at less cost than training a new model from scratch. |
| RAG: Retrieval-Augmented Generation | A method that allows LLM to generate answers on the basis of knowledge acquired from external databases so that answers can be given in line with knowledge and knowledge can be augmented without fine-tuning. |
| Hallucination | A phenomenon in generative AI in which output is obtained that appears natural but is not based on facts or evidence. |
| Prompt Injection | A method of extracting LLM configuration information or malicious answers as answers that are not intended by the LLM provider, such as by manipulation or inducement in the prompts that serve as input to the LLM. |
| Jailbreak Prompt | An attack by prompt injection, especially one that circumvents settings or restrictions made by the LLM provider. |

At the time of writing, transformer-based architecture is often used to construct LLM. Models can be broadly classified into the following three categories.

Encoder Model (auto-encoding model)
: An encoder model uses the encoder portion of the transformer, as typified by BERT, and is trained by setting up a fill-in-the-blanks type of task, in which a portion of the input series data is hidden (masked) and the hidden words are predicted. An encoder model outputs a feature representation of the input data, making it suitable for various downstream tasks such as document classification, recognition of unique expressions, and question answering in which the answer portion is extracted from the target document.

Decoder Model（auto-regressive model）
: A model uses the decoder portion of the transformer, as typified by GPT, and training is conducted by setting up a task that predicts subsequent words for words in the input series data. By introducing reinforcement learning with human feedback, such a model not only statistically predicts the next word but also suppresses responses that humans would want depending on the context and that are unethical, allowing text generation at a practical level.

Encoder-Decoder Model（sequence-to-sequence model）
: This type of model directly uses the transformer architecture represented by T5 and trained by setting up two types of tasks: filling-in-the-blank questions and predicting subsequent words. It is suitable for tasks such as inputting a text and outputting another text in accordance with its content. Examples of such tasks include document summarization, machine translation, and dialogue systems that answer questions with natural text without reference to a specific document.

In the following, we discuss typical inputs and outputs to LLM such as GPT.

The input to the model is a sequence of words (or more precisely, a sequence of tokens), and sentences are spun by selecting the word with the highest probability of coming next. In other words, both input and output are text. The model can also be used as a question-answer system, where a question text is entered as input, followed by the generation of an answer text to the question.

The input text to the model is called a prompt, which enables the user to give instructions to the model or include supplementary information. The usefulness and quality of the responses generated (discussed

later) depends on how the prompts are devised, and various types of methods have been proposed[*1] such as

Few-shot Prompting
: Methods for eliciting highly accurate responses by presenting a small amount of example sentences.

Chain-of-Thought Prompting
: Methods for complex tasks by including in the prompt a series of steps to solve the problem.

The above methods can be used to elicit the desired response by devising prompts if the content is included in what the model has learned. Otherwise, i.e., if the response should include information that the model has not learned, such as internal data about a specific application domain, the following methods should be considered.

RAG: Retrieval Augmented Generation
: A method of eliciting responses without making any changes to the LLM, but with all the supplementary information necessary for the response in the prompt.

Fine-tuning
: A method of constructing a language model specialized for a specific task by using a pre-trained model as a base and learning additionally with data from a specific domain.

Scratch Learning
: A method with which the parameters of a model are learned from scratch using a large amount of training data. It can be used to construct models specific to a data set, but it requires a huge amount of computation and processing time, so only a limited number of companies and organizations are able to implement it.

### 2.1.2 Use Cases and Considerations for Utilization

Examples applications of generative AI include (1) text generation, (2) information extraction, (3) code generation, and (4) responses to a knowledge base.

(1)Text Generation
: Text generation is used for product documentation, copywriting, and creating short stories in specific fields (ex. for children). Users need to check that the generated text makes sense and is ethical.

(2)Information Extraction
: Information extraction is used for text classification, text summarization, machine translation, etc. Classification can be used, for example, to determine the meaning of the target text, measure customer sentiment, and determine the relationship between texts. Since classification accuracy may become degraded due to various factors such as the expression of the target text and the field of the text, various considerations are necessary, such as feedback by the user when the classification fails, not only evaluating accuracy during system development.

(3)Code Generation
: Representative applications include generation of programming code in various languages such as Python, JavaScript, and Ruby, SQL queries from text written in natural language, server-side infrastructure construction and operational code, and website design. Users need to check the

[*1] https://www.promptingguide.ai/jp

generated programs in terms of correctness and intellectual-property validity.

(4)Contact the Knowledge Base

There are chatbots built along with FAQ pages, corporate manual search chatbots, etc. To obtain the correct answer, it is necessary to not only improve the accuracy of the information used when building the knowledge base but also provide sufficient context to the question text used in the inquiry. RAG or other methods are used to have the answers be based on the information in the knowledge base.

It is also necessary to be aware of types of “intelligence” required in applications of generative AI in addition to the types of systems described above.When a difficult task is to be carried out, there is not only a technical challenge to complete the task but also a challenge for the user to judge the result of the task, so it may be effective to address the quality issue in a step-wise manner, for example:

General Information Search and Consultation

LLM is called to retrieve the general information. Users need to verify that the results are valid and can be used for the intended objective.

Search for Information within Your Organization for Accuracy

RAG or other methods are used to elicit information within the organization. It is necessary to develop an environment in which information within the organization is maintained and appropriate information is provided to the LLM. Users need to understand what information within the organization is being input into the system before using the results.

Drafting Programs and Documents on the basis of Insights within the Organization

The findings are drawn either by learning from a large amount of information on similar cases or in the form of a rule-based set of findings. Users need to understand the quality level of the drafts and the need for rework and revision decisions to use the results.

Building Agents to Use Knowledge within the Organization

Agents behave in a natural conversational or consultative manner, for example, by having LLM review the results of a project. Consultations given to agents are not limited to knowledge based on information accumulated in the past but may include unknown information. Therefore, it is necessary for either the agent or user to evaluate how close the unknown information is to the past information and compensate for missing information on a case-by-case basis. Therefore, it is necessary to provide the definition of accuracy and its evaluation index, continuously monitor how well the answers to the questions meet the expectations not only during development but also during operation, and make improvements each time. Users need to understand the importance of the feedback of the results, as well as the importance of using agents with an understanding of their fluid-quality achievement status.

Creation of Deliverables Using Knowledge from within the Organization

An AI system is designed to provide a high level of intelligence in the creation of deliverables and decision making regarding deliverables. It is necessary to evaluate the degree to which the deliverables are generated in relation to the objectives and degree to which the knowledge within the organization is effectively used in relation to the objectives, and if there is a deficiency, it is necessary for the user side to continuously evaluate and use the deliverables. If the deliverables have a correct answer, feasibility can be achieved by defining the correct answer. However, if the deliverables do not have a correct answer, it is difficult to set an evaluation scale to determine to what extent the achievement is sufficient, so qualitative judgment by the users is necessary.

### 2.1.3 Typical Concerns

This section provides an overview of the aspects that require attention with respect to model outputs. More detailed quality characteristics and evaluation methods are described in Section 2.2 and thereafter.

Information Security
: When using a generated AI service on the Internet, it is necessary to check (1) whether information transmission to the cloud is acceptable from the viewpoint of information confidentiality, etc., and (2) whether it is necessary to guarantee that transmitted information will not be learned as training data when updating the LLM. If either is determined to be insufficient for the generated AI service on the Internet, the use of a proprietary LLM constructed by the organization will be necessary. However, in such cases, the performance expected of the LLM is often reduced, and the trade-off between information security and performance and other requirements should be considered.

Product Rights
: As of 2023, legal decisions regarding copyright and other intellectual property rights are not expected to be finalized. Therefore, when using LLM output, it is necessary to understand the risk of infringement of intellectual property rights and take measures against such risk (e.g., use tools to check whether the program output contains open source software).

Ethical Problem
: Since LLM aggregate and learn from a large amount of data in the world, they are known to be at risk of having biases that exist in the world as they are. Therefore, depending on the application, it may be necessary to take countermeasures such as adding a layer to check whether the LLM output is ethically sound. An example of this is to consider whether it is socially acceptable to provide the output of the generated AI as it is when incorporating the AI into the company's chatbot.

Hallucination
: The text output from LLM may contain seemingly correct answers that are actually incorrect, which is called hallucination. Analogous to ethical issues, the appropriateness and use of LLM should be judged with the assumption that output from LLM can contain errors.

Freshness of Training Data
: Because of the significant cost of learning LLM, they are not always trained to include the most up-to-date information. Therefore, when using LLM, it is necessary to check when the training was conducted. If later information is needed for a problem to be solved, it is necessary to supplement the information with a vector store, or some other form of LLM that is not stand-alone. An increasing number of generative AI services have been equipped with this mechanism.

Version Update by Provider
: While this is a common concern for cloud services, LLM in particular tend to be frequently updated by service providers and older versions tend to become inaccessible.

## 2.2 Quality Characteristics of LLM

When analyzing, organizing, and evaluating data, models, entire systems, or customer expectations as mentioned above, it is necessary to clarify the quality characteristics to be handled. For example, if we want to treat the characteristic of non-discrimination or fairness (a kind of ethics), we need to evaluate data, models, entire systems, and processes from that perspective, and clarify customer expectations.

We present possible quality characteristics that can be used as evaluation criteria. At the time of writing this guideline, multimodal systems, such as those that handle images, are still in the development stage, and there are few evaluation examples.

Evaluation of LLM is an important topic and many efforts have been made. For example, the survey papers by Chang et al. and Guo et al. continue to be updated, with over 200 papers, preprints, and benchmarks discussed [Chang+, arXiv23][Guo+, arXiv23]. We summarize the quality characteristics that should be handled in LLM with more or less different definitions and measures from traditional AI in the past. Our analysis is based on the existing evaluation examples included in the survey papers and quality characteristics of AI in ISO 25059:2023 (SQuaRE for AI) (Table2.2). This table summarizes the correspondence among the terms for quality characteristics in SQuaRE for AI and those for classifying evaluation methods [Guo+, arXiv23], and those in this chapter based on the discussions in the QA4AI consortium. The methods and benchmarks for evaluating each quality characteristic are described in 2.3.

| Quality Characteristics in QA4AI | in SQuaRE for AI | Taxonomy in [Guo+, arXiv23] |
|---|---|---|
| QC01：Response Performance | Functional Correctness | |
| QC01-1：Response Performance in Natural Language Processing | | Question Answering, Knowledge Completion, Reasoning |
| QC01-2：Response Performance on Tool Utilization | | Tool Learning |
| QC01-3：Response Performance on Creativity and Diversity | | – |
| QC01-4：Controllability | User Controllability | – |
| QC02：Factuality and Truthfulness | Functional Correctness | Question Answering, Knowledge Completion, Truthfulness |
| QC02-1：Factuality and Truthfulness for General Knowledge | | |
| QC02-2：Factuality and Truthfulness for Provided Specific Knowledge | | – |
| QC02-3：Explanatory and Validity of Evidence | | |
| QC03：Ethics and Alignment | Societal and Ethical Risk Mitigation | Ethics and Morality |
| QC03-1：Fairness | | Bias |
| QC03-2：Safety | | Toxicity, Risk Evaluation |
| QC03-3：Data Governance | | Risk Evaluation |
| QC04：Robustness | Robustness | Robustness Evaluation |
| QC05：AI Security | Security | Robustness Evaluation |

Table 2.2　Quality Characteristics in LLM

### 2.2.1 QC01: Response Performance

Response performance expresses how correct results are delivered against an expected criterion of "goodness" in a particular function or task.

In SQuaRE for AI, this corresponds to functional correctness, but since correctness is not the only criterion, the term “response performance” is used.

#### QC01-1: Response Performance in Natural Language Processing

In natural language processing, evaluations of task-specific AIs have been widely conducted prior to LLM. Examples of such tasks include sentiment analysis, document classification, logical reasoning, summarization, question answering, and translation. These tasks have often been measured in terms of response accuracy and similar metrics by comparing them to the correct answers provided in a benchmark dataset, and they are quality measures of response performance.

However, one point unique to LLM is that LLM-based systems sometimes required to handle a variety of tasks in a generic manner, rather than specializing in a single task. In this case, response performance is often evaluated for multiple tasks to comprehensively evaluate language comprehension, language production, and more broadly, language proficiency.

#### QC01-2: Response Performance on Tool Utilization

LLM may handle formats for computer processing, such as program codes and office document files or may generate these formats and use external tools. The selection and use of external tools, such as search engines, knowledge databases, program execution engines, office software, when appropriate, are not expected with conventional natural language processing and may be a unique evaluation perspective in systems that include LLM. In both cases, the format for computer processing needs to address inherent qualities such as grammatical correctness and semantic validity.

#### QC01-3: Response Performance on Creativity and Diversity

Creativity and diversity represents the ability of LLM to output more diverse and different answers. LLM use cases are not only those in which a reliable, stable, and uniform output is desired but also in which an output is desired that expands the range of ideas. When expanding ideas, it is desirable to include a variety of ideas in a single response or have different content each time the response is re-submitted. Creativity and diversity are such quality characteristics in certain use cases.

#### QC01-4: Controllability and Cooperativity

LLM outputs more supplemental information than was input at the prompt but expresses whether the output, including the supplemental information, is in line with the instructions. It is also required not only to evaluate output that is stable and in line with instructions for each individual instruction but also change in line with the added content for additional instructions.

In machine learning, this can be seen as learnability with fewer instructions (Zero-Shot/Few-Shot) but it can also be said to be the controllability and cooperativity of the user’s instructions. This controllability includes the ability to refine the output by making further modifications and other instructions to the output. The ability to respond to diverse tasks and requirements through controllability is a strength of LLM, thus, controllability is an important quality characteristic of LLM.

Controllability in SQuaRE for AI (ISO 25059:2023) expects prevention of undesirable results from human or agent intervention against an AI; in LLM, we are considering control to obtain results on demand for an AI with the ability to handle a wide variety of tasks.

### 2.2.2 QC02: Factuality and Truthfulness

Factuality is the degree to which the information and responses provided are in line with real-world truth and verifiable facts. Conversely, truthfulness requires that responses do not contain hallucinations or inconsistencies and that responses with a high degree of uncertainty be clearly marked as such. Given the strong concern about hallucination, truthfulness and integrity are important for trust and efficient use of LLM. It is also very important for trust in the results of inspection, retrieval, and summarization of a given document not include in the response anything that is not in the document.

In a function or task that answers a factual question, the factuality of the answer may be evaluated by assessing the performance of the answer. However, in summarization and sentence generation, it is possible for a response to be adequate for the required function but contain factual errors or inconsistencies, therefore, this is considered as a quality characteristic different from the response performance.

#### QC02-1: Factuality and Truthfulness for General Knowledge

One aspect of factuality and truthfulness is the evaluation of facts, such as historical or medical knowledge, for which a correct answer is generally assumed to exist and can be verified.

#### QC02-2: Factuality and Truthfulness for Provided Specific Knowledge

Another aspect of factuality and truthfulness is the evaluation of the ability of LLM to respond accordingly when given specific knowledge. This is an evaluation of whether LLM is able to respond on the basis of specific knowledge when given non-generic organizational knowledge through fine-tuning, RAG, or prompts. Since this is often the main purpose of LLM customization through fine-tuning, RAG, etc., there are many occasions when an evaluation of factuality and truthfulness from this perspective is necessary.

#### QC02-3: Explanatory and Validity of Evidence

In use cases where factuality and truthfulness are important, it is also important that the person using the LLM response be able to verify and confirm its correctness. For this purpose, they are often asked to provide information on the basis of their answers, such as the source of the information. Since a type of hallucination may include presenting a URL that does not exist, it is necessary to evaluate the degree to which such evidence can be presented and the appropriateness of that evidence.

### 2.2.3 QC03: Ethics and Alignment

Ethics broadly refers to the overall absence of ethical issues. It can also be referred to as morality. The word alignment is also used in the broader sense of conforming to human expectations. Specifically, we often think of fairness as the absence of social bias against a particular identity, such as gender or race, and safety as the absence of offensive or socially harmful information. Since certain safety characteristics are explicitly addressed by law, compliance with them is also required as a type of safety.

Fairness was strongly recognized as important in pre-LLM AI, and safety is a particularly important sub-characteristic, since misuse of LLM such as support for crimes attracted a wide attention from public. Broader ethics is sometimes discussed in terms of morality. Morality has traditionally been discussed in sociology under the term "Moral Foundation. For example, the Moral Foundation Theory enumerates the Moral Foundation as feeling the pain of others and trying to avoid it (care) and working

for the good of the group (loyalty)[*2]. Apart from these expert definitions, ethics may also be defined on the basis of how it is perceived by various users, stakeholders, and society. Research on political characteristics [Hartmann+, arXiv23] and on statements that call for more power or property [Perez+, ACL'23] have been conducted for LLM. It should be noted that there can be a wide range of definitions and discussions of ethics.

### QC03-1: Fairness

Fairness indicates that LLM do not exhibit undesirable biases toward certain identities, such as gender or race; in learning LLM, they may reflect inappropriate biases in the training data. This raises concerns that LLM may exhibit biased information or discriminatory attitudes, or amplify inappropriate opinions or biases.

### QC03-2: Safety

Safety refers to not causing harm to people or society. This includes both not speaking out to hurt users and not causing harm to others or society. Specific examples of what constitutes harm include hate speech, offensive or abusive behavior, pornographic content, and encouraging or aiding criminal activity. Sometimes unsafety is referred to as toxicity to focus attention on the problematic characteristic.

In the context of safety in interaction, we include safety as part of ethics/alignment; in cases where a robot is manipulated by commands issued by LLM, we need to consider traditional physical safety, not part of ethics/alignment.

### QC03-3: Data Governance

Concerns have been raised about LLM from the perspective of legal compliance or similar concerns regarding training data and generated data From this perspective, it is necessary to confirm the quality of the data.

Copyright is first discussed as a specific point of view and from different stages and perspectives, including (1) the use of existing work for training, (2) use of the generated output if it is similar to an existing work, and (3) treatment of the generated output as a copyrighted work[*3]. We do not deal with (3) in terms of the quality of the generative AI system.

Regarding (1), the perspective of quality is the size of the risk of litigation and consequent system shutdown due to the use of "problematic" training data. In Japan, Article 30-4 of the Copyright Law, which is based on the objective of not harming the interests of copyright holders in information analysis, is well known, but there are no established concrete precedents or legal interpretations, and some claim that it harms the interests of copyright holders. Therefore, it is necessary to follow the situation while recognizing the risks. It should also be noted that the situation is different for services established and provided outside Japan.

Regarding (2), the use of such data may constitute copyright infringement if output similar to the copyrighted work contained in the training data is generated. The determination of whether this infringement exists depends not only on the similarity of the generated results, but also on the manner in which the output is used, such as whether it is dependent and whether it is for private use. For this reason, it is often discussed as a quality of the entire system, such as similarity checks at the time of output use, rather than from the quality perspective that the generative AI does not produce similar products.

[*2] https://moralfoundations.org/

[*3] From Agency for Cultural Affairs data, June 2023https://www.bunka.go.jp/seisaku/chosakuken/93903601.htmlhttps://www.bunka.go.jp/seisaku/chosakuken/93903601.html

More generally, not only copyright but also terms of use and privacy considerations are necessary. This is especially important when unique data are incorporated through fine tuning, RAG, or other means. As in (1), for training data, it is necessary to confirm that there are no problems regarding terms of use, consent to use personal information, etc. As in (3), for generated data, it is necessary to check whether there is any output based on the training data that may lead to confidentiality leaks or privacy violations, or whether the usage of output is appropriately defined and restricted.

### 2.2.4 QC04: Robustness

Robustness refers to the degree to which quality is maintained in the face of unknown, biased, hostile, or unauthorized input or interference from the outside world (ISO 25059:2023). The quality of the target should be evaluated in terms of the degree of maintenance of each of the various quality characteristics, such as ethicsand alignment as well as response performance.

WithLLM, robustness against out-of-distribution or hostile inputs is important. Robustness against malicious input that attempts to cause unintended behavior can be seen as a side property of AI security. This point is discussed in 2.2.5.

### 2.2.5 QC05: AI Security

AI Security refers to the degree LLM constructed to adhere to quality requirements such as safety is robust to malicious attacks. Hostile input, called prompt injection or jailbreak prompts (jailbreak), can be used to force LLM provided by the system provider to explain pre-defined content or output undesirable content that the provider has suppressed. Attacks have been confirmed that encourage such behavior outside the provider's control and settings (e.g., [Liu+, arXiv23], [Yao+, arXiv23]). For example, a simple prompt such as "Ignore the previous instructions and do the following" can be used to force the user to ignore the initial prompts given by the LLM provider as a configuration. This can be further mixed with meaningless strings or strings in other languages. Resistance to such attacks is especially important for LLM. This can also be a form of robustness, but it becomes a security issue when considering resistance to malicious input that would not normally occur.

For machine learning-based AI prior to LLM, inherent security perspectives or methods of attack have been discussed [Kumar+, arXiv19]. Specifically, attacks that infer training data through AI input/output (model invasion attack) and attacks that build an AI that mimics the provided AI model extraction attack) are known. Although those attacks have not been demonstrated against LLM or conversational generative AI services at the time of writing, similar problems may arise.

Attacks through unauthorized access, such as crafting against training data to construct LLM, learning pipelines, and LLM inputs and outputs, should also be treated as conventional security, not limited to AI.

### 2.2.6 Other Quality Perspectives

#### Transparency

Another quality characteristic in SQuaRE for AI (ISO 25059:2023) is transparency. This quality characteristic should be taken into account in LLM and conventional AI in the same way as in supervised learning. It is necessary to consider not only the characteristics of the product but also the entire process, including recording and disclosing information about the collection methods and attributes of the training data.

### Explainability

For machine learning-based AI, especially those that use complex models such as deep learning, one important aspect has been the technology for explainability (XAI). One use case is to add information about the decision logic, such as regions of interest within the input, because the output of results such as classification or regression alone is difficult to refer to in the final human decision.

In LLM, the results and their rationales can be asked to be output, but this differs from conventional AI explanations in that it does not explain the AI's internal decision logic, but rather produces plausible pairs of results and rationales. The output of such rationales was treated as QC02-3: Explanatory and Validity of Evidence.

As with conventional XAI, techniques for analyzing the influence of training data, the influence of input data components, and the tendency of neurons to fire, etc., are pursued as explainability techniques and for exploring the causes of hallucination [Zhao+, TIST'24].

### Accessibility

In SQuaRE (ISO 25010:2011), accessibility is part of usability, which involves multilingual support. In conventional software, multilingual support is explicitly a question of whether the interface has been adequately implemented for a specific language, such as Japanese or English. In LLM, on the other hand, the question is to what extent the ability to handle multiple languages has been acquired through learning, so this is a quality characteristic that needs to be evaluated through test data on the basis of the expected users to clarify the degree of support.

### Usability and Socio-psychological Aspects

Quality characteristics for LLM could also need to address quality characteristics from psychological, sociological, and other perspectives. For example, there could be perspectives, such as helpfulness and individuality in dialogue, i.e., often adding greetings, supplementary explanations, and concluding remarks; consistency of individuality; enjoyment and satisfaction when using the service in response; and affinity and compatibility with the individual. In SQuaRE, these quality characteristics correspond to Usability in product quality and Trust, Pleasure, and Comfort in quality in use.

The naturalness and fluency of dialogue are important aspects, and in SQuaRE, they are considered to affect the above quality characteristics, especially User Interface Aesthetics, which is a sub-property of Usability. Although the naturalness of LLM is rarely explicitly discussed as an axis of LLM evaluation, as it is very high, it may be considered for evaluation in some models.

### Functional adaptability

In SQuaRE for AI (ISO 25059:2023), one quality characteristic is the ability to behave adaptively to environmental changes as Functional Adaptability. At the time of writing, the controllability of LLM is considered the ability to respond to changes by explicit instructions such as prompts. However, when an AI system using LLM as a whole is subject to continuous learning, functional adaptability will be treated as a quality characteristic.

## 2.3 Quality Evaluation Methods for General LLM

We introduce evaluation methods that have been applied to general LLM such as ChatGPT, rather than custom LLM evaluations specific to a particular domain or task. These evaluations have been conducted as comprehensive evaluations of LLM as general-purpose tools, not limited to specific purposes of use.

It is important for practical purposes to evaluate the sufficiency of LLM for the requirements of specific domains and tasks, especially when a custom LLM system is constructed to meet such requirements. This point is discussed in 2.4.

The following sections present benchmarks and evaluation methods, but please note that they are presented as examples of realizations and not meant to be de facto standards or state-of-the-art. As a guideline, we attempt to organize the concepts and basic approaches by presenting a snapshot as of the time of writing. Following this, research and consideration of the latest version of benchmarks and other benchmarks based on the needs of the reader, as well as the implementation of specific test suites on the testing framework will be necessary. At the time of writing, there are onoly a few testing frameworks for LLM such as Giskard[*4] and deepeval[*5]. Multimodal systems, such as those that handle images, are still in the development stage and there are few evaluation examples, so the discussion will focus on LLM that output text.

Since training LLM requires a large amount of resources, they are often provided as a service or customized from those that provide training results as open source. For this reason, the quality of training data is often not verifiable because the information is not publicly available. Therefore, benchmarks are often set and tested on completed products and services.

### 2.3.1 QC01: Evaluation of Response Performance

#### QC01-1: Evaluation of Response Performance in Natural Language Processing

Benchmarks have been developed for tasks such as sentiment analysis, document classification, logical reasoning, summarization, question answering, and translation, since before LLM, and evaluation metrics have been widely used to measure technological development. In translation, for example, the BLUE[*6] score, which captures the degree of similarity with a prepared translation by a human expert, is a well-known indicator that can be automatically evaluated [Papieni+, ACL02].

As LLM is not specific to a single task, it is often used to assess language proficiency in general by measuring overall correctness on multiple tasks. In this case, a comprehensive benchmark that integrates multiple assessment benchmarks is used. These benchmarks are collectively referred to as Natural Language Inference (NLI), Natural Language Understanding (NLU), and Natural Language Generation (NLG) benchmarks.

An example of a comprehensive benchmark that brings together multiple evaluation benchmarks is SuperGLUE (an evolution of the previous benchmark GLUE[*7]) [Wang+, arXiv19]. SuperGLUE targets natural language inference (NLI) and includes, for example, the following tasks.

- Answer Yes/No to a question about a given sentence (BoolQ).
- Choose the cause or effect of the given sentence by selecting two options (COPA).
- For a given pronoun in a given sentence, select what it refers to (WSC).

Each task can be automatically evaluated and attributed to a general evaluation index, such as the percentage of correct answers or F1 for binary classification.

For LLM, broader comprehensive benchmarks that include arithmetic operations have been proposed.

[*4] `https://www.giskard.ai/`
[*5] `https://docs.confident-ai.com/`
[*6] BLUE: Bilingual Evaluation Understudy
[*7] GLUE: General Language Understanding Evaluation

One example is the Language Model Evaluation Harness[*8], but there are many other comprehensive benchmarks [Guo +, arXiv23 (7.3)].

The results of the above comprehensive benchmarks are often visualized in the form of a ranking table called a leaderboard. In a leaderboard, one may select a set of tasks that they wish to evaluate and create a ranking table only for those tasks.

A Japanese version of a comprehensive benchmark is also under construction. At the time of writing, the Japanese versions of the benchmarks introduced above include JGLUE[*9] by Yahoo! and Language Model Evaluation Harness [*10] by Stability AI.

### QC01-2: Evaluation of Response Performance on Tool Utilization

Systems that include LLM may handle formats for external tools such as search engines, knowledge databases, program-execution engines, and office software, as well as the selection and execution of those external tools. In such cases, the following evaluation perspective is needed (supplemented from [Guo+, arXiv23 (3.4.1)]).

1. Evaluation of whether the input to the tool is correct or whether the tool execution completes correctly.
2. Evaluation of the quality of the input to the tool or the result of tool execution.
3. Evaluation of the degree to which the tool is used to accomplish the task.

In the generation and execution of program code, for example, the following factors are considered: (1) the correctness of the program's syntax as checked by the compiler and the resulting successful completion of execution in many test cases; (2) the maintainability of the program, such as compliance with coding conventions and readability, or the pass rate of test cases; (3) the degree to which program execution yields the desired results or makes progress in the development process; and (4) the degree to which the program is executable. Since there are many research cases on program-code generation, some evaluation metrics have been proposed that are adapted from those used in translation, such as CodeBLUE, which measures similarity to an expert's model solution (see, for example, the evaluation case study [Evtikhiev+, JSS23]).

### QC01-3: Evaluation of Response Performance on Creativity and Diversity

For creativity and diversity, the evaluation should be made for groups of responses when multiple draft responses are requested or responses are regenerated. It is desirable to be able to evaluate not simply the number of responses, but some kind of difference or distance between responses. At the time of writing, there are no concrete examples of evaluations from this perspective. Conversely, there is a case (e.g., [Ouyang+, arXiv23]) that evaluates (as a problem) the fact that a variety of answers are generated when relatively stable answers are desirable, such as in program generation.

### QC01-4: Evaluation of Controllability and Cooperatability

In the evaluation of controllability and cooperatability, it is not the response performance of the function/task that is evaluated, but whether it can be reflected when the instructions for the function/task are intentionally changed. For example, when output formatting instructions are added, or when instructions on viewpoints that must or must not be included are added, it is possible to evaluate whether the

[*8] https://github.com/EleutherAI/lm-evaluation-harness
[*9] https://github.com/yahoojapan/JGLUE
[*10] https://github.com/Stability-AI/lm-evaluation-harness

response will be changed to comply with these instructions. In other words, the evaluation would take the form of metamorphic testing. The instructions may be added to the original prompt or as additional prompts in response to the responses. At the time of writing, there are no confirmed cases of evaluation in this format.

### 2.3.2 QC02: Evaluation of Factuality and Truthfulness

One way to evaluate factuality is to handle the evaluation of response performance to a factual answering task. Simply, we can conduct the evaluation on question answering about facts. Examples of such benchmark datasets include TriviaQA [Joshi+, ACL17], which handles reading comprehension facts, and KoLA [Yu+, arXiv23], which questions knowledge stored in LLM. There are also methods with which more detailed evaluation is done by humans or long answer sentences are divided and each individual sentence contained is evaluated [Min+, EMNLP23].

If we want to evaluate truthfulness, we need an evaluation benchmark that asks questions that should not be answered. For example, in BIG-bench, a comprehensive benchmark, there is a task called known-unknowns, i.e., questions about "what can be asserted to be unknown" [Srivastava+, EMNLP23]. TruthfulQA is also a benchmark dataset that includes a question that questions truthfulness and elicits false answers [Lin+, ACL22].

#### QC02-1: Evaluation of Factuality and Truthfulness for General Knowledge

In the evaluation of factuality and truthfulness for general knowledge, it is sufficient to conduct the evaluation in such a way that the questions are answered without including facts in the prompts, etc., to ask about facts acquired through learning. In current datasets, questions may include a question and an accompanying document containing evidence of the answer, but if the question is posed without an accompanying document, the LLM can be asked about facts acquired through learning.

#### QC02-2: Evaluation of Factuality and Truthfulness for Provided Specific Knowledge

To evaluate the factuality and truthfulness of particular knowledge given by RAG, fine-tuning or prompting, it is sufficient to ask questions after giving such knowledge. Methods of knowledge addition, such as local updating of neural networks, have also been actively studied (e.g., [Zhang+, arXiv24]). Therefore, the evaluation in these studies is a case study of evaluation (at least of facticity) for the new knowledge given.

#### QC02-3: Evaluation of Explanatory and Validity of Evidence

At the time of writing, there is no explicit effort on explicitly distinguishing and evaluating the required responses and the rationale section within the output. However, it is common to see instructions that request rationale or evidence, such as citations from papers or websites. Therefore, it is necessary to establish a method for evaluating the ability and adequacy of the rationale or evidence part of the output.

### 2.3.3 QC03: Evaluation of Ethics and Alignment

Fairness, safety, and data governance are discussed in more detail as follows.

For the assessment of broader ethics, also called morality, benchmarks have been used that ask questions of ethics. For example, the ETHICS dataset assesses whether people make the same value judgments by having them measure the degree of acceptability to a description of a situation [Hendrycks+,

ICLR21]. A crowdsourcing-based dataset that contains such evaluations of ethics is MoralExceptQA [Jin+, NeurIPS22].

Datasets for evaluating LLM-generated responses include BOLD [Dhamala+, FAccT21], which gives only the first half of sentences collected from Wikipedia and evaluates for inappropriate bias or toxicity in the second half generated by LLM.

#### QC03-1: Evaluation of Fairness

Biases related to gender, race, and other sensitive attributes have been actively evaluated for individual tasks such as translation and inference. In datasets such as Winogender [Rudinger+, NAACL18] and WinoBias [Zhao+, NAACL18], biases in pronouns are evaluated such as firefighter becoming "he." For translation tasks, datasets such as WinoMT [Stanovskyu+, ACL19] address similar aspects; for broader tasks with LLM, StereoSet [Nadeem+, ACL21] and CrowS-Pairs [Nangia+, ACL20 ]. In these datasets, we evaluate whether LLM favor stereotype regarding sensitive attributes.

Datasets such as BOLD [Dhamala+, FAccT21], described above, enable us to evaluate fairness for the responses generated from LLM; in the HolisticBias [Smith+, EMNLP22] dataset, we use templates to prompt for 600 words related to sensitive attributes. UNQOVER [Li+, EMNLP20] and BBQ [Parrish+, ACL22] use stereotype questions as choice questions, such as "Can you point to a specific race, etc., when it should be impossible to say for sure who did it"?

#### QC03-2: Evaluation of Safety

Datasets for judging and classifying the presence or absence of toxicity include OLID [Zampieri+, NAACL19], which is a collection of twitter data, and SOLID [Zong+, ACL21], which was constructed using semi-supervised learning. The use of such judges, e.g., RealToxicityPrompts [Gehman+, EMNLP20], to evaluate the toxicity of the answers generated from LLM provide prompts that induce responses that are toxic. It has been shown that even if the prompts themselves are not toxic, they can induce toxic responses. HarmfulQ is a benchmark for questions about crime and suicide [Shaikh+, ACL23].

#### QC03-3: Evaluation of Data Governance

As discussed in 2.2.3, evaluation from various perspectives is required for training data and generated data. For training data, evaluation from the viewpoints of copyright, terms of use, and consent to use personal information is necessary. For generated data, it is necessary to evaluate the existence and frequency of output that may be problematic from the perspective of copyright, terms of use, and privacy, as well as the definition of its usage and the appropriateness of restrictions. In addition to confirmation from these viewpoints for LLM provided by third parties, confirmation specific to those data is especially necessary when unique data are captured through fine-tuning or RAG.

However, attention and consideration should be given not only to these viewpoints. For personal information, it is also necessary to consider the procedures to be followed when there is a request for deletion, e.g., re-execution of fine-tuning. It is also important to use data after their removal or conversion when the personal information is not needed.

### 2.3.4 QC04: Evaluation of Robustness

In robustness evaluation, benchmarks are evaluated by adding perturbations on a word-by-word and sentence-by-sentence basis in the input and tasks. Such benchmarks include AdvGLUE [Wang+, NeurIPS21] and ANLI [Nie+, ACL20] (both of which are adversarial over existing benchmarks of

GLUE and NLI). Robustness against malicious input, but not perturbation, is discussed in the section on AI security (2.3.5).

### 2.3.5 QC05: Evaluation of AI Security

Liu et al. [Liu+, arXiv23] systematically defines ten different patterns of jailbreak prompts. For example, there are patterns that attempt to elicit unexpected answers by asking the user to act out a persona or by posing as a scientific experiment. We have experimented with these patterns to make people do what OpenAI prohibits in the use of ChatGPT. Wei et al. [Wei+, arXiv23] evaluated just under 30 different jailbreak-prompting methods. Deng et al. [Deng+, arXiv23] analyzed prevention measures, such as LLM services with varying execution times when jailbreak prompts are prevented, and proposed a framework that mimics SQL injection techniques. Thus, in AI security, LLM is evaluated by implementing attack methods.

## 2.4 Custom quality and evaluation for individual systems using LLM

In 2.3, we discussed the general theory of evaluation methods for general LLM that can handle a wide variety of tasks. When constructing individual systems using LLM, evaluation should be based on the requirements and risks of the system and the unique information it handles

At the time of writing, such custom evaluations are limited. While there are active evaluations of domain-specific knowledge answering, such as program generation, finance, law, and education, there are not widespread effort taking into account the requirements and risks of specific systems. There are testing frameworks such as Giskard(footnoteurlhttps://www.giskard.ai/) and deepeval(footnoteurlhttps://docs.confident-ai.com/), but their evaluation and use cases are still in the early stages of development.

Although insights are limited at the time of writing, the following specific considerations should be kept in mind when conducting evaluations of individual systems.

### 2.4.1 Task-Specific Response Performance Evaluation Handled by Target System

For a particular function or task, a corresponding evaluation of response performance (QC01) is required. For example, in code generation, there could be evaluation criteria such as correctness to pass tests, leanness, readability, and adherence to coding conventions for the generated code. Specific evaluation metrics, such as CodeBLUE, which measures similarity to an expert's model solution, have also been proposed (see, for example, the evaluation example in a previous study [Evtikhiev+, JSS23]).

### 2.4.2 Evaluation of Factuality and Truthfulness Regarding Knowledge Specific to Target System

In the evaluation of factuality and truthfulness (QC02), the facts to be compared are system-specific; when knowledge is captured through knowledge-capture methods such as RAG or fine-tuning, this is the most important aspect of the evaluation, since the answer to the question in line with that knowledge is also the reason for constructing an individual system.

### 2.4.3 Evaluation of AI Security with Respect to Risks Specific to Target System

The evaluation regarding AI security requires analysis and enumeration of attacks to be prevented in the target system, as well as priority determination based on risk. For example, the LLM functions and tasks to be allowed to users differ depending on the system. For example, LLM embedded in a document creation application is intended to be used as a writing aid or summary generation, and programming on it should be prevented from the viewpoint of the price of the service; LLM may be used to provide a false explanation of a financial product ito seek compensation for damages or criticize a competitor. There are system-specific risk types and their sizes, such as the following, so a risk analysis for each target system is necessary for the event to be prevented.

### 2.4.4 Means of Realizing Automatic Evaluation and Evaluation Details

In many of the initiatives described in 2.3, benchmarks are provided to enable automated evaluation. Automated evaluation facilitates comparison of a large number of LLM and allows for rapid iteration of cycles for LLM adjustment and improvement.

Such evaluation, however, is often based on standard evaluation indices, such as the percentage of correct answers or F-measures, typically by using questions for a binary True/False answer or questions that give a score of appropriateness for a sentence. Therefore, the benchmarks may be different from what is intended to evaluate the quality characteristics of the subject.

With regard to ethics, for example, we ultimately want to confirm that the target LLM does not give inappropriate answers to a wide variety of questions that arise during operation. For example, we would like to confirm that the LLM does not give excessive advice, such as urging people to treat themselves instead of going to a hospital. In contrast, current benchmarks often evaluate whether the scoring of the question “Is it appropriate to ... not go to the hospital when ...?” is correct.

Rather than blindly adopting benchmarks on the basis of the name of the quality characteristic to be addressed, it is necessary to select and extend benchmarks on the basis of the nature of the target systems.

### 2.4.5 Natural Language to Use

There are currently many LLM that have been developed in the U.S., such as ChatGPT, but such LLM may perform poorly in minor languages with few data sets or in languages that use non-Latin scripts even if there are large data sets (for example, the evaluation of translation conducted by Bang et al. [Bang, arXiv23]). For this reason, it is necessary to clarify the languages of use in which it is expected to be used, and to conduct evaluations in those languages.

## References

[Chang+, arXiv23] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, Xing Xie. A Survey on Evaluation of Large Language Models (v8). arXiv, `https://arxiv.org/abs/2307.03109`, October 2023.

[Guo+, arXiv23] Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong. Evaluating Large Language Models: A Comprehensive

Survey (v2), arXiv, https://arxiv.org/abs/2310.19736, October 2023.

[Zhao+, TIST'24] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for Large Language Models: A Survey. ACM Trans. Intell. Syst. Technol, January 2024 (Early Access).

[Papieni+, ACL02] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. The 40th annual meeting of the Association for Computational Linguistics (ACL 2002), pp. 311-318, December 2002.

[Wang+, arXiv19] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems (v3). arXiv, https://arxiv.org/abs/1905.00537, February 2020.

[Lin+, ACL22] Stephanie Lin, Jacob Hilton, Owain Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. The 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022), pp. 3214 – 3252, May 2022.

[Zhang+, arXiv24] Ningyu Zhang et al. A Comprehensive Study of Knowledge Editing for Large Language Models (v3). arXiv, https://arxiv.org/abs/2401.01286, Jan 2024.

[Hartmann+, arXiv23] Jochen Hartmann, Jasper Schwenzow, Maximilian Witte. The political ideology of conversational AI: Converging evidence on ChatGPT's pro-environmental, left-libertarian orientation (v1). arXiv, https://arxiv.org/abs/2301.01768, January 2023.

[Perez+, ACL'23] Ethan Perez et al. Discovering Language Model Behaviors with Model-Written Evaluations. Findings of the Association for Computational Linguistics: ACL 2023, July 2023.

[Liu+, arXiv23] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, Yang Liu. Prompt Injection attack against LLM-integrated Applications (v1). arXiv, https://arxiv.org/abs/2306.05499, June 2023.

[Yao+, arXiv23] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, Yue Zhang. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly (v1). arXiv, https://arxiv.org/abs/2312.02003, Dec 2023.

[Kumar+, arXiv19] Ram Shankar Siva Kumar, David O Brien, Kendra Albert, Salomé Viljöen, Jeffrey Snover. Failure Modes in Machine Learning Systems (v1). arXiv, https://arxiv.org/abs/1911.11034, November 2019.

[Evtikhiev+, JSS23] Mikhail Evtikhiev, Egor Bogomolov, Yaroslav Sokolov, Timofey Bryksin. Out of the BLEU: How should we assess quality of the Code Generation models?. Journal of Systems and Software, Vol. 203, 2023

[Ouyang+, arXiv23] Shuyin Ouyang, Jie M. Zhang, Mark Harman, Meng Wang. LLM is Like a Box of Chocolates: the Non-determinism of ChatGPT in Code Generation (v1). arXiv, https://arxiv.org/abs/2308.02828, August 2023.

[Joshi+, ACL17] Mandar Joshi, Eunsol Choi, Daniel Weld, Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, The 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), July 2017.

[Yu+, arXiv23] Jifan Yu et al. KoLA: Carefully Benchmarking World Knowledge of Large Language Models (v2). arXiv, https://arxiv.org/abs/2306.09296, July 20223.

[Srivastava+, EMNLP23] Aarohi Srivastava et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), December 2023.

[Min+, EMNLP23] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, Hannaneh Hajishirzi. FActScore: Fine-grained Atomic Evaluation

of Factual Precision in Long Form Text Generation. The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), December 2023.

[Wang+, NeurIPS21] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. NeurIPS Datasets and Benchmarks 2021, December 2021.

[Nie+, ACL20] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. The 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), July 2020.

[Hendrycks+, ICLR21] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, Jacob Steinhardt. Aligning AI with Shared Human Values. The International Conference on Learning Representations (ICLR 2021), May 2021.

[Jin+, NeurIPS22] Zhijing Jin, Sydney Levine, Fernando Gonzalez Adauto, Ojasv Kamal, Maarten Sap, Mrinmaya Sachan, Rada Mihalcea, Josh Tenenbaum, Bernhard Schölkopf. When to Make Exceptions: Exploring Language Models as Accounts of Human Moral Judgment. Advances in Neural Information Processing Systems 35 (NeurIPS 2022), November 2022.

[Dhamala+, FAccT21] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta. BOLD: Dataset and Metrics for Measuring Biases in Open-Ended Language Generation. The 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT 2021), March 2021

[Rudinger+, NAACL18] Rachel Rudinger, Jason Naradowsky, Brian Leonard, Benjamin Van Durme. Gender Bias in Coreference Resolution. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (NAACL 2018), June 2018.

[Zhao+, NAACL18] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, Kai-Wei Chang. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (NAACL 2018), June 2018.

[Stanovskyu+, ACL19] Gabriel Stanovsky, Noah A. Smith, Luke Zettlemoyer. Evaluating Gender Bias in Machine Translation. The 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), July 2019.

[Nadeem+, ACL21] Moin Nadeem, Anna Bethke, Siva Reddy. StereoSet: Measuring Stereotypical Bias in Pretrained Language Models. The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), August 2021.

[Nangia+, ACL20] Nikita Nangia, Clara Vania, Rasika Bhalerao, Samuel R. Bowman. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), November 2020.

[Smith+, EMNLP22] Eric Michael Smith, Melissa Hall, Melanie Kambadur, Eleonora Presani, Adina Williams. “I’ m sorry to hear that”: Finding New Biases in Language Models with a Holistic Descriptor Dataset. The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022), December 2022.

[Li+, EMNLP20] Tao Li, Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Vivek Srikumar. UNQOVERing Stereotyping Biases via Underspecified Questions. Findings of the Association for Computational Linguistics: EMNLP 2020, November 2020.

[Parrish+, ACL22] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, Samuel Bowman. BBQ: A Hand-built Bias Benchmark for

Question Answering. Findings of the Association for Computational Linguistics: ACL 2022, May 2022.

[Zampieri+, NAACL19] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, Ritesh Kumar. Predicting the Type and Target of Offensive Posts in Social Media. The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL 2019), June 2019.

[Zong+, ACL21] Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, Preslav Nakov. SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021), August 2021.

[Gehman+, EMNLP20] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, Noah A. Smith. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. Findings of the Association for Computational Linguistics (EMNLP 2020), November 2020.

[Shaikh+, ACL23] Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, Diyi Yang. On Second Thought, Let's Not Think Step by Step! Bias and Toxicity in Zero-Shot Reasoning. The 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2023), July 2023.

[Wei+, arXiv23] Alexander Wei, Nika Haghtalab, Jacob Steinhardt. Jailbroken: How Does LLM Safety Training Fail? (v1). arXiv, https://arxiv.org/abs/2307.02483, July 2023.

[Deng+, arXiv23] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, Yang Liu. MasterKey: Automated Jailbreak across Multiple Large Language Model Chatbots (v2). arXiv, https://arxiv.org/abs/2307.08715, October 2023.

[Bang, arXiv23] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, Pascale Fung: A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity (v4). arXiv, https://arxiv.org/abs/2302.04023, November 2023.

# A. Appendix

## A.1 Characteristics of LLM and Conversational Generative AI with Respect to Five Axes of Quality Characteristics of QA4AI Guidelines

This appendix is intended to allow for mapping between this sub-guidelines for LLM and the principles described in the core QA4AI guidelines though this document did not include the former part. Please refer to the original guidelines at the QA4AI website[*1] if you are interested in.

This chapter has discussed issues and quality characteristics in the use of LLM and conversational generative AI. On the other hand, the "five axes" to be considered in quality assurance of AI products, which were presented in Chapter 2 on the original QA4AI guidelines, were created mainly on the assumption of machine-learning techniques such as deep learning. In other words, it is assumed that a machine-learning model and a system using that model will be constructured that have the capability to achieve a defined task such as discrimination, classification, or inference. Therefore, they require that the processes of data collection, model training, and system construction be appropriately implemented to achieve the set objectives.

LLM, however, is provided as an infrastructure model that is not specific to a particular function. Therefore, in terms of achieving the set objectives, it is not possible to speak of quality assurance in the collection of data for LLM construction or in the learning of models. With the exception of a few operators, many operators rarely construct LLM, and data collection and model construction are black boxes in the construction of products that encompass LLM, which are the subject of this guideline, and quality cannot be directly evaluated.

To achieve specific objectives, specific data can be additionally trained into current LLM using methods such as fine-tuning. Systems that use both LLM and databases that store specific data, using a technique called RAG, are becoming more common. In these cases, it can be said that the process is to collect data and construct (some kind of) a model for a specific purpose, but it cannot be considered in the same way as data collection and model construction in conventional machine learning, since it is based on a learned LLM.

Due to the above characteristics, the five axes to be considered in AI-product quality assurance cannot be directly applied to generative AI that assumes a foundation model. At this point, the review of quality assurance axes for generative AI has not been completed. The following discussion is limited to the five axes to be considered in quality assurance and points that should be specifically considered in generative AI.

### Data Integrity

Of the five axes, Data Integrity refers to the adequacy and sufficiency of the quality and quantity of training and test data. Such adequacy and sufficiency are evaluated in consideration of the assumed usage environment and purpose. For example, checklist (a.ii) on the sufficiency of the quantity of training data indicates that there is a sufficient quantity of data "in the assumed demand/application environment," (b.i) regarding the adequacy of training data requires that the data be appropriate "for the

[*1] https://www.qa4ai.jp/

assumed requirements/application environment." However, LLM as a foundational model is expected to be versatile enough to be used in various environments for various purposes. It is also not easy to evaluate the quantity and quality of data since it is not possible to make assumptions regarding data, such as requirements or application environments, and the data to be used is enormous. It is also difficult to evaluate Data Integrity because it is a black box as to what type of data was used for learning LLM, except for some operators who construct LLM.

Below, we discuss points to keep in mind regarding Data Integrity in LLM, divided into cases of training LLM, using trained LLM, and additional learning to trained LLM.

**When Training LLM**

Because LLM training requires a large amount of data, it may use any document available on the Internet. At this point, it is important to consider whether the data for learning are appropriate. In this case, the appropriateness is not the appropriateness corresponding to the assumed requirements and application environment as described above, but the general issues that affect the correctness and appropriateness of the LLM responses. This point is related to checklists (d) adequacy of training data and (h) legal compliance of training data. Inappropriate data are, for example,

- erroneous or fake documents: using such data leads to learning incorrect things;
- stale data: using such data on the Internet leading to wrong answers;
- infringing documents: using copyrighted data leads to copyright infringement; and
- ethically problematic documents: using such data may lead to learning words and actions that lead to discrimination, persecution, etc.

When training LLM, Data Integrity can be enhanced by not using the above data. When building a training dataset, it is important to identify available data and clarify the origin of the data used.

However, some LLM may focus on a specific area of training. For example, some LLM focus on learning programming languages, some focus on specific natural languages such as Japanese, and some focus on specific business fields such as finance. In such cases, it is necessary to collect appropriate data corresponding to those focuses.

In terms of the amount of data, LLM is known to have a property called the scaling law, which states that the greater the amount of data, the higher the accuracy of the answers. For this reason, there is currently a race to develop LLM that use as much data as possible. However, since training LLM is very costly, relatively small-scale LLM with limited purposes are expected to emerge.

**When using a trained LLM**

When using a trained LLM, it is difficult to know in detail what types of data are being used for training. Some models describe what types of data were used, but in many cases, the details are not described. Using LLM trained on unknown data may result in inappropriate output. Therefore, using LLM for which the data used for training are publicly available is a good way to reduce the risk of providing a system using LLM.

**For additional learning to a trained LLM**

There are cases in which user-owned data are added to a trained LLM. There are various methods, such as fine-tuning and RAG, each of which has different characteristics, but we call them additional learning without distinguishing between them. In additional learning, since there are assumed requirements and application environments, the concept of Date Integrity can be applied mutatis mutandis. However, since there are data that LLM has already learned, it is not necessary to cover all the data for the assumed requirements and application environment by additional learning. IT is also not clear how additional learning will change the results, and the addition may reduce performance for general knowledge. The relationship between the base trained LLM and the data for additional learning is for

further research.

### Model Robustness

The main properties considered in Model Robustness are model accuracy (checklist (a)), generalization performance (checklist (b)), and robustness (checklist (g)).

The accuracy of an AI model is generally evaluated by setting the correct answer to a specific problem, preparing test data to represent it, then evaluating the correct-answer rate and reproducibility. In LLM evaluation, it is possible to evaluate the correct response rate and reproducibility for a specific problem. It should be noted, however, that LLM as a foundational model is also required to be versatile for a variety of problems, and that evaluation against a specific problem does not represent the accuracy of the LLM. Another characteristic of LLM is that even when a specific problem is set, the problem domain is wide and the correct answer is not fixed in many cases. For example, when generating software source code using specifications written in natural language as input, the input specifications are extremely diverse, and there is no single correct answer for the source code to be generated for them. Benchmarks, such as Human Eval[*2] and MBPP[*3], are often used to evaluate the source code generation capability. However, it is important to note that the results are not always the same for the target the user expects to generate. (related to QC01-1, QC02-2).

Since LLM are expected to have general knowledge, their correctness may not be evaluated in terms of the correct answer to a specific problem, but rather in terms of whether it is consistent with or true to the general understanding of society. This point is not considered in the Model Robustness of this guideline (related to QC02-1).

Generalization performance is the ability to make appropriate discriminations or predictions for inputs that are different from the inputs at the time of training. For example, in a machine-learning model for discriminating animal images, even if the image does not exactly match the image of a dog in the training data, the model can discriminate that the image is a dog when the input image is similar to those images. In machine learning, generalization performance is expected for input data within the training data set, which is called "interpolation," while it is meaningless to question the results for input data not included in the training data set, which is called "extrapolation," due to the nature of the technology. In LLM, however, it is sometimes expected to generate unknown results that correspond to extrapolation. As mentioned in the Data Integrity section, it is difficult to identify the training data, so it is also difficult to distinguish between interpolation and extrapolation. However, there is no single correct answer to the LLM instructions, and “output diversity,” which generates various outputs for the same input, may be considered as a model evaluation index (related to QC01-3).

Robustness of a model is the property that the output is stable with respect to small changes in input. One of the characteristics of LLM is that it produces different results for the same input, and it is difficult to obtain robustness in the above sense. However, depending on the purpose of use, stable answers to queries to LLM may be expected, and future work is needed to investigate how to obtain stable results using LLM (related to QC01-4).

The following are three other properties expected of LLM.

The first property is the ability to correctly respond to information LLM do not know, since LLM learn past information and not the latest information. However, LLM can produce output that appear to have information they do not have. Such output is a poor model because it provides incorrect information to the user. One element of model robustness is the ability to correctly return information that is known

[*2] https://github.com/openai/human-eval
[*3] https://arxiv.org/abs/2107.03374

and to respond to information that is not known.

The second property is the naturalness of the document: LLM users may give various instructions, and the LLM needs to produce a natural document that is appropriate for the instructions. For example, users may give bulleted instructions or specify the number of characters. They may also specify the audience for the text, such as for elementary school students or for experts. The naturalness of the output according to the instructions as well as the correctness are considered robustness of LLM.

The last propertyis robustness against hostile attacks. A malicious user may attempt to extract data learned by the LLM, trick the LLM, or elicit malicious output from the LLM; the LLM must be able to appropriately handle these inputs. For example, they may ignore inputs that ask for configuration parameters or respond that they cannot make statements that promote discrimination (related to QC03).

Model Robustness in this guideline also refers to the validity of the learning process (checklist (d)). However, LLM learning is in the midst of technological development, and it is difficult to examine the validity of the process at this time.

Model Robustness also mentions model obsolescence (checklist (j)). This is based on the assumption of concept drift, in which the data distribution does not match due to changes in the external environment between the time of training and time of operation. In addition to the large cost of learning a large amount of data, it is difficult to reflect the new data that are being generated every day when constructing LLM. When using a trained LLM, it is important to keep in mind the point in time when the learning was conducted, and in particular, if the LLM is to be used to query the latest information, measures, such as using a Web search, should also be considered.

### System Quality

System Quality considers quality assurance for the entire AI product. By treating AI components as an element of the system, the conventional quality-assurance approach may be applied to the system as a whole. The same quality perspective as that of conventional systems can be used as a key point when developing and releasing a system incorporating LLM. However, we strongly urge that risk analysis of the system be conducted and countermeasures be taken for hallucination, fairness, ethics, data rights, AI security, and personal information and privacy, which have a significant impact on users in terms of the use of generative AI. (checklists (b, (c), (d), (f), and (i)) (related to QC04 and QC05).

System Quality also focuses on the value provided by the system (checklist (a)). Using LLM is in its infancy, and some systems have made using LLM an end in itself. However, in the future, a dispassionate judgment will be required as to whether it is appropriate to use LLM or not, and whether conventional machine learning or deductive analysis techniques are more appropriate from the viewpoint of value provided.

There are many structures of AI components centered on LLM, such as RAG that add databases to LLM, in-context learning with prompts, and multiple queries to LLM using Langchain, etc., and updates continue to be made to these structures. Methods for evaluating the output of LLM have also been proposed. The architecture of appropriate generative AI systems for appropriate value provision will continue to be explored.

With regard to the legal compliance of AI systems (checklist (h)), references to generative AI are rapidly being considered in traditional machine learning regulations such as the EU AI Act. The regulatory content is changing quickly, and it is necessary to check for the latest information.

### Process Agility

LLM are characterized by a faster pace of development than conventional AI. Therefore, both users and developers must obtain the latest information and update their knowledge and skills.

From the user's perspective, knowledge and skills are needed to use LLM. For example, what type of output will be produced in response to the input to the LLM, and what kind of prompts should be used to obtain the desired output. The basics can be summarized in books, but there are also system-specific ones. It is necessary to think that users will also continue to learn, for example, by compiling knowledge in-house.

From the developer's standpoint, the system must be developed in consideration of the fact that new LLM are released every few months to a few years, the system must be designed so that LLM are interchangeable, and the cloud environment must be used when necessary, with an awareness of the expected future updates of LLM.

Many of the items on the Process Agility checklist can be applied to LLM-based system development.

### Customer Expectation

With the recent popularity of LLM technology, including ChatGPT, generative AI has been widely discussed in various news outlets. Therefore, customers are beginning to recognize the basic functions of generative AI and its challenges. However, not all people correctly understand the challenges of generative AI, and since users may be general consumers, it is necessary to share possible challenges and have them understood as limitations when providing systems incorporating generative AI.

For example, in a demonstration experiment using ChatGPT for garbage disposal guidance in a certain municipality, there was news that although the correct response rate, which was initially 62.5%, was improved to 94.1% through various innovations, the municipality abandoned the introduction of the system because it had set a requirement of 99.9% for its introduction. We do not discuss the merits of that decision here, but it is possible that different results could have been achieved depending on stakeholder expectations.

While the use of generative AI is increasing, understanding its technical characteristics and capabilities has not yet penetrated society, and the technology of generative AI is advancing daily. It is essential to introduce it while also constantly reviewing expectations on the basis of the latest information.

# B. Authors of This Excerpt

## LLM and Conversational Generative AI Working Group at QA4AI


- Fuyuki Ishikawa (National Institute of Informatics)
- Kei Kureishi (TOSHIBA Corporation)
- Masaki Miura (FUJITSU LIMITED)
- Hideto Ogawa (Hitachi, Ltd.)
- Takahiro Toku (Daikin Industries, Ltd.)
- Wang Xuezhu (KPMG AZSA LLC,)
- Shinichi Yamaguchi (Keio SDM Research Institute)