

Quality Metrics for LLM-Generated Asset Administration Shells: A Perturbation-Based Evaluation Approach

Janek Groß¹[0000–0002–6306–711X], Elena Zentgraf¹[0009–0008–1031–2709], and Jens Heidrich¹[0000–0001–6967–4722]

University of Applied Sciences Mainz, Mainz, Rhineland-Palatinate, Germany

Abstract. The rapid digital transformation of manufacturing, often referred to as Industry 4.0, relies on seamless interoperability between physical and software assets. A central enabler is the Asset Administration Shell (AAS), a standardized digital representation of such assets. Recent advances in large language models (LLMs) enable the generation of AAS submodels from unstructured sources such as product datasheets but raise challenges for quality assurance. In particular, unexpected errors, the lack of ground truth references, and the absence of standardized quality metrics hinder reliable adoption. In this work, we evaluate quality metrics for AI-generated AAS using a perturbation-based evaluation framework. By systematically degrading AAS generation along multiple dimensions, we assess how well different metrics reflect quality changes. Based on a dataset of 200 products from multiple manufacturers, we generate 6,400 AAS instances using GPT-4o-mini, Qwen3, and DeepSeek-R1. Our results show that metrics based on exact matching of property names and similarity-based soft matching of property values, in particular value-based recall and name-based F1 score, provide the most reliable indicators of quality degradation. Furthermore, we quantify the impact of different perturbation types and analyze differences across model families and product segments. These findings support the selection of suitable metrics, the tuning of LLM-based pipelines, and the integration of AI-generated AAS into industrial applications.

Keywords: Quality Metrics · Evaluation · Asset Administration Shell · Large Language Models · Information Extraction · Digital Twins · Industry 4.0

1 Introduction

Asset Administration Shells (AAS) [5, 6] provide standardized digital representations of hardware and software assets (see Fig. 1). They define common semantics for properties and services and enable interoperable digital twins in Industry 4.0. As such, a large number of AAS must be created to cover the variety of assets used in modern manufacturing.

Typically, manufacturers generate AAS from internal product data. However, if company-foreign or legacy products need to be modeled, manual extraction of

properties from datasheets or technical documentation is required. This process is time-consuming, error-prone, and requires domain expertise.

Information extraction (IE) is the task to derive structured, machine-readable information from unstructured sources such as tables or natural-language text [4, 2]. In the context of AAS creation, IE enables systematic extraction of product properties from technical documents. Traditionally, IE systems required extensive domain-specific annotation and complex pipelines. Recent advances in large language models (LLMs), particularly instruction tuning and function calling, have significantly lowered this barrier. LLMs now enable the construction of IE pipelines without supervised training, leading to emerging LLM-based AAS generation tools [22, 19, 10]. While these tools are promising, they remain unreliable and often produce unexpected results, raising critical questions about output validity and quality.



AAS "Festo_DPDM-Q-32-10-PA_99920200616214520000010828" V1.0	
▷ SM "Nameplate"	[http://smart.festo.com/id/instance/99920200616214520000010828]
◀ SM "TechnicalData" V1.0	[http://smart.festo.com/id/instance/99920200616214520000010828]
Prop "Stroke"	= 10 [mm]
Prop "Piston_diameter"	= 32 [mm]
Prop "Assembly_position"	= Any
Submodel Element (Property)	
Referable:	
idShort:	Stroke
description:	[en] Stroke
Semantic ID:	
semanticId:	(GlobalReference) https://admin-shell.io/DataSpecificationTemplates/DataSpecificationTemplate/1.0/Stroke
ConceptDescription	
Referable:	
idShort:	Stroke
Data Specification Content IEC61360:	
preferredName:	[en] stroke
shortName:	[en] Stroke
unit:	mm
definition:	[en] maximum travel of a fluidic linear drive
Property	
valueType:	xs:double
value:	10

Fig. 1. Example AAS [7] with three technical properties and property metadata.

To enable trustworthy use of LLM-based tools for AAS generation, suitable quality metrics are necessary. Such metrics serve both the evaluation and optimization of generation tools including choice of model, prompting strategies and other hyperparameters. Defining suitable quality metrics, however, is non-trivial. AAS content is often non-unique: the same product can be described using different classification systems (e.g., ECLASS, ETIM, or company-specific schemes). Moreover, variations in naming, alternative units, or permissible structure com-

plicate the application of standard IE metrics such as precision, recall, and F1 score. Furthermore, LLM-based generation performs multiple IE steps—such as entity recognition, normalization, and template mapping—in a single prompt-response cycle which prevents step-by-step evaluation. Together with the scarcity of labeled datasets, this underscores the need for tailored evaluation approaches.

In this work, we propose an empirical methodology for the evaluation of AAS quality metrics based on controlled perturbations of the generation process. By systematically degrading AAS instances, we analyze how well different metrics reflect changes in quality. Our research addresses the following questions: (1) Which metrics best reflect AAS quality degradation under controlled perturbations? (2) How do perturbation types and model characteristics affect AAS generation quality? (3) How does AAS quality vary across manufacturers and product segments?

Our contributions are (i) a set of AAS-specific quality metrics, including full-reference and no-reference variants, (ii) a perturbation-based evaluation methodology for systematically assessing metric performance, and (iii) an empirical analysis of AAS quality across perturbation types, model families, manufacturers, and product segments. These contributions support the selection and benchmarking of AAS generation tools, enable automated quality assurance, and facilitate the integration of AI-assistants into industrial processes.

The remainder of this paper is structured as follows. Section 2 examines relevant background and related work. In Section 3, the study design, including the evaluated metrics, the perturbation-based evaluation approach, and the experimental setup is described. Section 4 presents empirical results. Section 5 discusses the implications of the findings. Section 6 outlines threats to validity. Lastly, Section 7 concludes the paper and highlights directions for future work.

2 Background and Related Work

This section reviews relevant work on the evaluation of information extraction and LLM-based systems.

Traditional software quality models such as ISO/IEC 25010 [8, 20] provide valuable frameworks for assessing software products based on characteristics such as functionality, reliability, and maintainability. However, these models primarily target conventional software and do not readily extend to structured artifacts such as AAS submodels. In contrast, our work focuses on the empirical evaluation of concrete quality metrics tailored to such AI-generated artifacts, rather than proposing an encompassing quality model.

The evaluation of generative models like LLMs poses a challenge due to the complexity of the output domain. It is common practice to evaluate LLMs on benchmark tasks with a unique, short answer to provide a general indication of model performance, but these benchmarks may not reflect performance in other, domain-specific tasks. For example, translation and summarization are commonly evaluated using metrics such as BLEU [15] and ROUGE [12], which rely on n-gram overlap between generated and reference texts. However, these

metrics are not well suited for structured outputs, where fields are short and multiple semantically equivalent representations may exist.

Information extraction (IE) research is closely related to the AAS generation use case. IE systems typically combine tasks such as named entity recognition (NER) [11], relation extraction (RE) [23], and slot filling [21], and are evaluated on labeled datasets using precision, recall, and F1 score. For AAS generation, however, labeled data for individual extraction steps is not available, and these IE steps (i.e. recognition of technical properties, value and unit normalization, mapping to AAS template) are often performed within a single LLM interaction. As a result, evaluation must be performed end-to-end and requires adaptations of classical metrics, such as soft matching for property names and values. These adaptations increasingly rely on AI-based methods like embedding-based similarity, to account for permissible variations and the absence of a unique correct output (e.g., color: gray vs. colour: grey).

Similar challenges arise in the evaluation of other multi-step LLM-based systems, such as retrieval-augmented generation (RAG) where output quality is difficult to assess directly. Recent work has therefore explored alternative evaluation strategies that closely align with our setting. These include indirect metrics based on downstream task performance and LLM-based judgments. These works also evaluate their methods through correlation with human judgments and test systems with intentionally varied retrieval quality.

For example, Sander and Dietz [18] assess RAG systems based on their ability to support follow-up questions, providing an indirect measure of output quality. Es et al. [3] propose RAGAS, an automated framework that uses LLM prompting and embeddings to evaluate relevance and faithfulness. Building on that, Saad-Falcon et al. [17] apply prediction-powered inference [1] to align automated evaluations with human judgments. To evaluate their approach, they simulate systems with controlled retrieval performance, which inspired the perturbation-based evaluation approach used in this work.

Overall, existing work highlights both the limitations of traditional evaluation methods for structured outputs and the need for more flexible, task-specific evaluation strategies. Building on insights from IE and RAG evaluation, our work introduces a framework for systematically benchmarking AAS quality metrics, even in the absence of high-quality labeled reference data.

3 Study Design

This study systematically evaluates quality metrics for AI-generated Asset Administration Shells (AAS) using a perturbation-based framework. We deliberately degrade AAS generation along multiple dimensions to analyze how sensitively and consistently different metrics reflect quality changes.

The overall evaluation pipeline is illustrated in Fig. 2. Starting from product datasheets and corresponding AAS, we construct structured prompts that combine the unstructured datasheet content with a predefined property dictionary derived from the reference AAS. This dictionary specifies the expected proper-

ties, including their names, value types, and units, and serves as a schema to guide the extraction process.

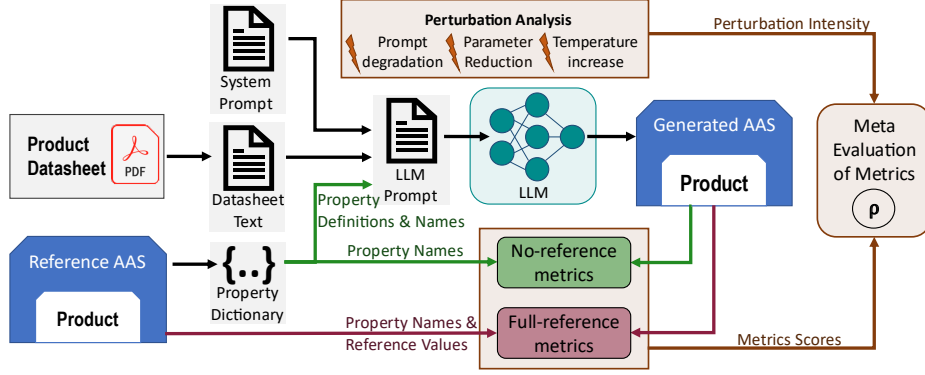


Fig. 2. Overview of the evaluation pipeline for AAS generation and metric-based quality assessment.

Based on these prompts, AAS instances are generated using LLM-based information extraction. The model is instructed to extract values for the specified properties and return the results in a structured format, which is then transformed into a technical data submodel of the AAS.

To simulate varying quality levels, controlled perturbations are introduced at different stages of the generation process, including prompt degradation, temperature variation, and the use of different model sizes. These perturbations systematically affect the completeness and correctness of the extracted properties.

The resulting AAS instances are evaluated using both full-reference and no-reference metrics. Full-reference metrics compare generated properties and values against the reference AAS, while no-reference metrics assess intrinsic plausibility, such as alignment with the expected property schema and the presence of missing values.

Finally, a meta-evaluation step analyzes the relationship between perturbation intensity and metric scores. By correlating metric values with controlled degradations, we assess how sensitively and consistently each metric reflects changes in AAS quality.

The following subsections describe the evaluated metrics, perturbation mechanisms, dataset, and experimental setup.

3.1 AAS Quality Metrics

We define a set of AAS-specific quality metrics for evaluating AI-generated AAS. These metrics focus on measurable differences in technical properties within

AAS submodels. Broader aspects such as schema compliance or adherence to AAS templates are not considered, as they can be addressed by deterministic validation tools.

We distinguish between two complementary types of metrics: **full-reference metrics** and **no-reference metrics**. Full-reference metrics compare a generated AAS against a reference and assess similarity in structure, properties, and values. While interpretable and reliable, such references are often unavailable in practice. No-reference metrics instead assess plausibility based on intrinsic characteristics, such as alignment with expected property names or missing values, making them suitable for large-scale quality assurance. Both types are therefore complementary.

To match generated properties to reference properties, we use a two-step procedure. First, similarity scores are computed for all property pairs using either normalized Levenshtein distance or cosine similarity of embeddings. Second, the Hungarian (linear sum assignment) algorithm determines an optimal one-to-one matching, ensuring that each generated property is matched at most once. A similarity threshold is applied to classify matches, optimized via grid search to maximize correlation with perturbation intensity.

For full-reference metrics, matched property values are additionally compared. Numeric values must fall within a tolerance threshold, while string values are evaluated using similarity thresholds.

Table 1. Overview of the evaluated metrics dimensions: reference type, matching type, and IE metric.

Dimension	Characteristics	Description
Reference Type	Name (no reference values required) Value (requires reference values)	Whether metric compares property names (against prompt) or property values (against reference AAS).
Matching Type	Exact (string equality) Embedding-based (cosine similarity) String-based (normalized-Levenshtein-distance-based similarity)	How property names are matched. For similarity-based matching, pairs match if similarity exceeds a threshold τ . Numeric values match if their relative difference is below 1%.
IE Metric	Precision Recall F1 Score	Standard information extraction metrics computed after the matching.

Overall, we evaluate 18 metrics covering all combinations of three dimensions: (i) reference type (names vs. values), (ii) matching type (exact, embedding-based, string-based), and (iii) information extraction metric (precision, recall, F1). Table 1 summarizes these dimensions. These metrics provide a simple yet effective basis for evaluating structured outputs, enabling both controlled benchmarking and scalable quality checks.

3.2 Perturbation-Based Evaluation Approach

To evaluate metric effectiveness, we introduce controlled perturbations during AAS generation to simulate varying quality levels. Metric sensitivity is then assessed by measuring the rank correlation (Spearman’s ρ) between perturbation intensity and metric scores. We consider three perturbation types:

LLM Temperature Variation The temperature parameter controls output randomness. Lower values produce deterministic outputs, while higher values increase diversity but typically degrade performance in logic tasks [16]. We exploit this effect to vary output quality.

Model Size Reduction Model size is reduced using smaller or distilled variants. While the functionality is largely preserved, performance degradation can be observed [9], enabling controlled quality variation.

Prompt Degradation The input prompt plays a central role in guiding the LLM and is therefore a potential leverage point for introducing controlled perturbations [14]. To analyze metric sensitivity, we apply a set of complementary perturbations at varying intensity levels (0.0–1.0). These perturbations introduce both syntactic and semantic distortions while preserving the overall task intent. The following perturbation types are used:

- **Semantic Drift:** Up to 10% of tokens are replaced with contextually plausible synonyms (WordNet [13]), introducing subtle shifts in meaning.
- **Grammar Degradation:** A portion of grammatical elements such as determiners, adpositions, auxiliary verbs, and punctuation are removed based on part-of-speech tagging, reducing syntactic clarity.
- **Information Ambiguity:** Hedge phrases (e.g., “maybe,” “or something”) are inserted at random positions (up to 10% of tokens), increasing ambiguity.
- **Contextual Irrelevance:** Grammatically correct but semantically unrelated sentences are inserted after up to 10% of sentences, introducing distracting context.
- **Lexical Noise:** Up to 10% of characters are modified (substitution, deletion, insertion, transposition) to simulate typographical errors.

All perturbations are scaled according to the specified intensity and applied jointly, resulting in a gradual transition from clean to heavily degraded prompts.

An example of prompt perturbation at maximum intensity (1.0) is shown in Table 2. The perturbed version is obtained by sequentially applying all perturbation types, resulting in a heavily distorted but still partially interpretable prompt. Despite severe degradation, LLMs often recover the task intent, highlighting both their robustness and the challenge of evaluating quality.

3.3 Dataset and Preprocessing

The evaluation is based on 200 products from four manufacturers (A–D), with 50 products each. Products were selected to maximize product diversity, while

Table 2. Original and perturbed prompt at maximum perturbation intensity (1.0).

Original Prompt	Perturbed Prompt (Intensity = 1.0)
You act as a text API to extract technical properties from a given datasheet. The datasheet will be surrounded by triple backticks (“”).	You cat textula matter API to estrat you know technical a liottle bti rope-jrtres given datasheet datasheet surrounded tyripebackticks

excluding incomplete or unusable data. Most products belong to the ECLASS segments “27: Electrical engineering” and “51: Fluid Power.” Table 3 summarizes dataset statistics.

AASX files were preprocessed to ensure compatibility with the basyx-python-sdk. This included normalizing deprecated URIs, correcting file references, merging duplicates, standardizing decimal formats, and sanitizing identifiers. Products were excluded if they used AAS version less than 3, lacked technical data submodels, or had unreadable datasheets. Additionally, only products with at least 10 purely technical properties not related to company or product identification were included.

To account for company-specific structures, a custom property dictionary was derived for each AAS, specifying property names, definitions, value types, and units. Table 4 summarizes the configurations.

The code to run the experiments and links to the product data is publicly available ¹.

Table 3. Data statistics. Number of technical properties per product and product coverage shown by the number of ECLASS categories (including broad product segments and increasingly detailed product groups, classes, and subclasses).

Company	Product count	Average property count (Std. Dev.)	Segment count	Group count	Class count	Subclass count
A	50	35.4 (12.3)	1	1	1	5
B	50	36.8 (4.8)	2	5	6	18
C	50	86.4 (33.9)	1	5	6	8
D	50	23.9 (9.7)	3	10	21	30
Overall	200	45.6 (30.6)	4	17	33	60

3.4 Models and Experimental Procedure

We evaluate both proprietary and open-source LLMs. Proprietary experiments use gpt-4o-mini via the OpenAI API, while open-source evaluations use Qwen3 (0.6B–32B) and DeepSeek-R1 (1.7B–70B) via a self-hosted Ollama deployment. All experiments were conducted on GPU-enabled nodes of an HPC cluster.

¹ <https://github.com/janek-gross/experiments>

Table 4. Fields used in the creation of custom property dictionaries for different manufacturers.

Company	Property Name	Definition	Value Type	Unit	Selection Constraints	ECLASS Release
A	id_short	description	string	None	>15 properties	11.0
B	display_name	concept_ description. description	value_type	unit*	ECLASS ids not ending in 90-99 >32 properties	12.0
C	preferred_name*	definition*	value_type	unit*	-	12.0
D	preferred_name*	definition*	value_type	unit*	>9 properties	14.0

*from the data_specification_content in the ConceptDescription.

The pipeline constructs prompts from datasheets and property dictionaries, instructing the model to extract property records. A json schema was used to structure the LLM output as a list of records with *name*, *value*, *unit* and *text reference* for each record, ensuring machine-readable results.

Generated AAS are evaluated using the defined metrics, and perturbations are applied to analyze metric sensitivity under controlled conditions.

4 Results

We first report data statistics to provide an overview of the generation results. In total, we generated 6,400 technical data submodels across 32 experimental conditions, covering different perturbation levels, model variants, and companies. Overall, 231,276 properties were extracted from 200 product datasheets.

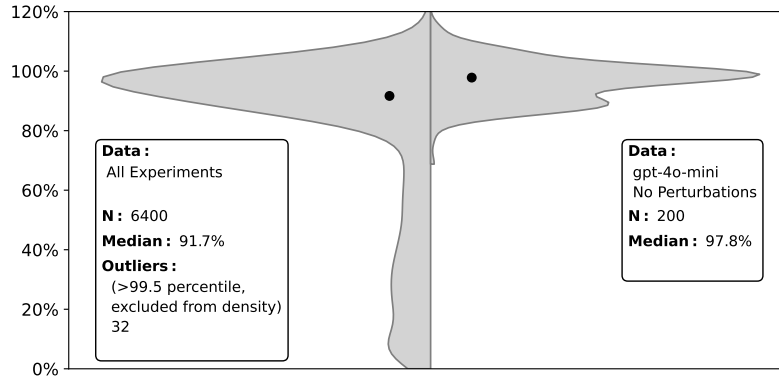


Fig. 3. Distribution of the ratio of extracted to prompted properties. Values larger than 100% indicate LLM-hallucinations or duplicate extractions. Values smaller than 100% indicate incomplete extractions.

On average, 83.5% of the prompted properties were extracted (correct or false) per submodel. In a baseline scenario without perturbations, gpt-4o-mini extracted values for 96.8% of prompted properties. Fig. 3 shows the distribution of the ratio of extracted to prompted properties. The left density includes all perturbations, while the right density represents an unperturbed baseline scenario.

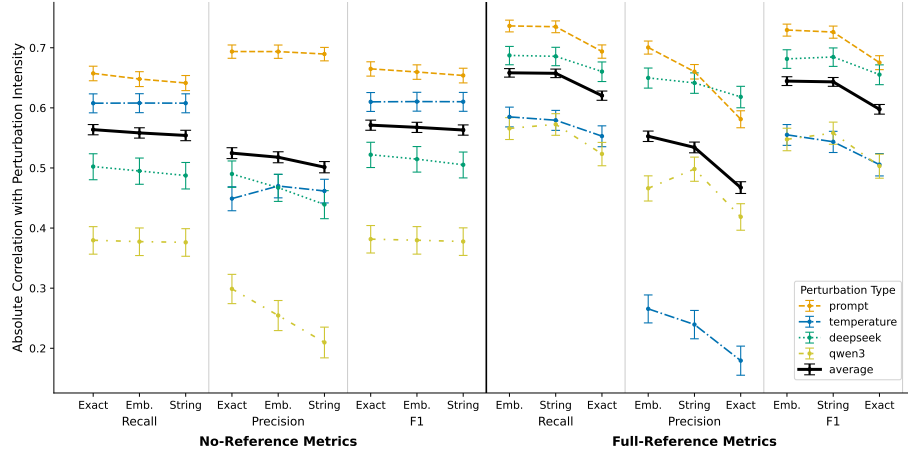


Fig. 4. Metric sensitivity measured as Spearman’s rank correlation with perturbation intensity. Error bars indicate the standard error of the correlation.

4.1 Metric Performance

We evaluate metric performance by analyzing the correlation between metric scores and perturbation intensity. Metric performance is assessed using Spearman’s rank correlation between metric values and the intensity of controlled perturbations (temperature, model size, and prompt degradation), addressing RQ1 and RQ2.

The results are shown in Fig. 4, which reports the absolute value of the Spearman correlation between metric scores and perturbation intensity for all combinations of reference type, matching strategy and information extraction metric (see Table 1). Higher values indicate greater sensitivity of a metric to quality degradation.

Several consistent trends emerge. In the no-reference setting, embedding-based matching performed better than string-based matching. However, neither provided a clear advantage over simple exact matching. For illustration purposes in Fig. 4, we therefore apply a consistent matching threshold of $\tau = 0.88$ to both string- and embedding-based no-reference metrics even though these metrics would otherwise default to exact matching with an optimal matching threshold

very close or equal to $\tau = 1.0$. **No-reference F1** scores exhibited the strongest monotonic relationships with perturbation intensity.

We utilize these observations and fix name-matching to exact matching in the full-reference setting to isolate the effect of value matching. In this scenario, soft matching based on embeddings resulted in the highest correlations. Among all candidates, **full-reference recall**, where cosine similarity was applied to values of exactly matching property names achieved the highest average correlation ($\rho = 0.65$) at a value matching threshold of $\tau = 0.88$. In both settings, precision-based metrics showed limited sensitivity to quality variations.

4.2 Impact of Perturbations

We next analyze the impact of perturbation types and model configurations on AAS generation quality, addressing RQ2. Fig. 5 illustrates the effects of different perturbations on the most sensitive metric full-reference recall, with perturbation intensity normalized between 0 and 1. Baseline performance for each model is indicated by markers on the left.

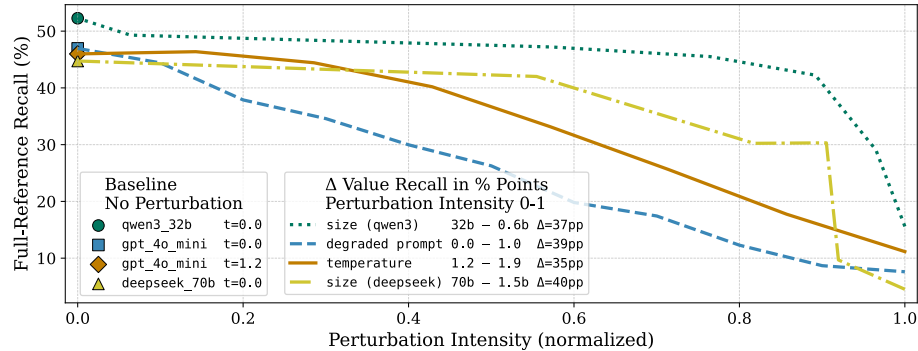


Fig. 5. Effect of perturbation types on value recall.

Across all perturbations, performance decreases by 35–40 percentage points, confirming that all perturbation types substantially degrade AAS quality. The effect of model size is distinctly non-linear: performance declines gradually for large models but drops sharply below approximately 8 billion parameters, suggesting a non-linear, potentially logarithmic relationship between model size and quality. This behavior further justifies the use of Spearman’s rank correlation, which captures monotonic but non-linear relationships.

In contrast, both temperature increases and prompt degradation show a more linear degradation trend across the tested ranges. Interestingly, in the baseline scenario, the open-source model Qwen3:32b achieves the best performance, indicating that architecture and training method can outweigh high parameter count alone.

4.3 Manufacturer and Product Segment Analysis

Finally, we analyze how AAS quality varies across manufacturers and product segments (RQ3). Fig. 6 compares both dimensions using strip plots overlaid with boxplots. The left plot shows differences between manufacturers, while the right plot highlights variation across product segments.

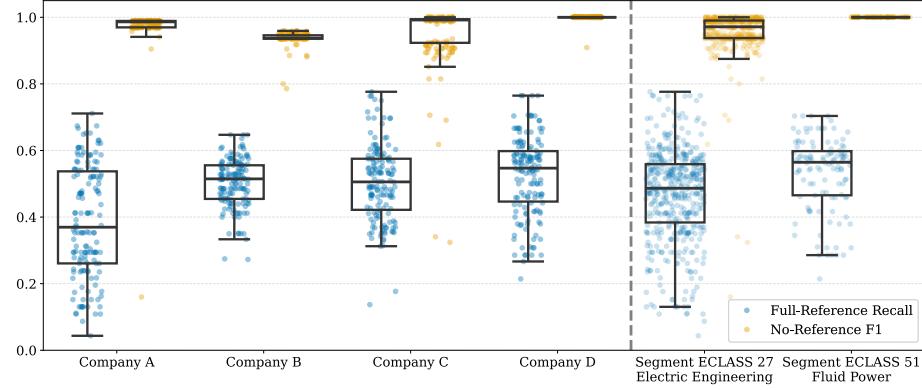


Fig. 6. AAS quality across manufacturers and product segments.

ANOVA indicates statistically significant differences between manufacturers for both no-reference and full-reference metrics ($p < 0.001$), with effect sizes of $\eta^2 \approx 0.13$ – 0.17 , suggesting that company affiliation explains a moderate portion of the observed variance. This indicates that product-specific characteristics have a stronger influence on AAS quality than manufacturer-specific factors.

At the segment level, Fluid Power products tend to achieve higher scores than Electrical Engineering products, although substantial variability remains within each segment. For no-reference metrics, a ceiling effect is observed, with many samples receiving near-perfect scores, reflecting that only indirect quality signals are captured in the absence of reference values.

5 Discussion

The results provide several insights into the evaluation of AI-generated AAS and the behavior of different quality metrics. A key finding is that for matching property names, simple, interpretable metrics based on exact matching outperform more complex similarity-based approaches. This indicates that exact matching more reliably captures quality degradation—such as missing or incorrectly named properties—while embedding-based approaches are intended to capture semantic similarity, they introduce noise in settings where precise and standardized terminology is required. However, for the comparison of property

values exact matching turns out too restrictive. Here, a soft matching based on similarity which permits spelling variations and semantically equivalent phrasing leads to the most sensitive metrics.

From a practical perspective, these findings are directly relevant for Industry 4.0 applications. In industrial pipelines, quality assessment must be transparent, reproducible, and easy to integrate. The identified metrics meet these requirements and can be used for benchmarking AAS generation tools, monitoring extraction quality, and supporting acceptance decisions prior to system integration. In particular, no-reference metrics enable scalable quality assurance in scenarios where reference AAS are unavailable.

At the same time, the results highlight limitations of metric-based evaluation. Optimizing a system with respect to a single metric can lead to metric overfitting, where models exploit metric-specific weaknesses rather than improving actual output quality. This is particularly relevant for no-reference metrics, which capture quality only indirectly and may exhibit ceiling effects, as observed in our experiments. In practice, robust evaluation therefore requires a combination of complementary metrics and, where necessary, human-in-the-loop validation.

The perturbation-based methodology itself provides additional insights. By systematically degrading inputs and analyzing metric responses, it enables controlled and reproducible assessment of metric sensitivity without requiring labeled ground truth data. This approach is not limited to AAS generation and can be applied to other structured information extraction tasks and LLM-based systems with end-to-end evaluation requirements.

Finally, differences across model families, manufacturers, and product segments indicate that AAS generation quality depends on both model characteristics and data properties. While larger models tend to perform more robustly, the results also show that architecture and training can outweigh parameter count. Variation between product segments further suggests that domain-specific factors, such as terminology consistency and the number and complexity of properties, significantly influence extraction performance. These findings highlight the importance of context-aware evaluation and caution against relying on single benchmark scenarios.

6 Threats to Validity

Our evaluation is subject to several threats to validity.

Construct Validity A central threat concerns the operationalization of AAS quality. In this work, quality is primarily measured through the correctness and completeness of extracted property names and values in technical data submodels. Although these measurements are important, they do not cover all aspects of AAS quality, such as appropriateness in a specific industrial context, compliance with domain-specific modeling conventions, or usefulness for downstream applications. In particular, proposed no-reference metrics measure plausibility only indirectly and fail to detect semantic errors.

Internal Validity Our perturbation-based methodology assumes that increasing perturbation intensity corresponds to decreasing AAS quality. Although this assumption is well motivated for the considered perturbation types, such as prompt degradation, temperature increase, and model size reduction, the relationship may not always be strictly monotonic for all models and products. Furthermore, the optimization of similarity thresholds based on correlation with perturbation intensity may bias the evaluation in favor of metrics that align particularly well with our perturbation design.

Conclusion Validity The conclusions drawn from the statistical analyses may be affected by variability in product characteristics, model behavior, and perturbation effects. While the dataset and number of generated AAS are substantial, some subgroup analyses, for example by manufacturer or product segment, are based on smaller effective sample sizes. Furthermore, Spearman’s rank correlation only measures monotonic relationships, but overlooks non-monotonic effects.

External Validity The generalizability of our findings is limited by the scope of the dataset and the evaluated models. Our experiments are based on 200 products from four manufacturers and focus on technical data submodels, with strong representation from specific ECLASS segments. The results may therefore not directly transfer to other industrial domains, other types of AAS submodels, or different product documentation styles. In addition, only a selected set of proprietary and open-source LLMs was evaluated, so the findings may not fully generalize to other model families or future model generations.

7 Conclusion and Future Work

In this work, we presented a systematic empirical framework for evaluating the quality of AI-generated Asset Administration Shells (AAS). By combining full-reference and no-reference metrics with a perturbation-based evaluation methodology, we enabled a controlled and reproducible analysis of how different metrics reflect variations in AAS quality. Our results show that metrics based on exact matching of names and similarity-based matching of values, provide the most reliable indicators of quality degradation across different perturbation types. These findings support the use of soft-matching techniques for benchmarking AAS generation tools in case of non-unique spelling and phrasing of property values.

Beyond the specific results, the proposed perturbation-based methodology provides a general approach for evaluating metrics in scenarios where labeled ground truth is scarce or unavailable. It enables systematic comparison of metrics under controlled conditions and can be transferred to other structured information extraction tasks and LLM-based systems.

In future work, we plan to extend this approach in several directions. First, we aim to incorporate expert assessments to better align automated metrics with human judgment and to calibrate metrics in realistic evaluation settings. Second, we

will investigate more advanced evaluation methods, including semantics-aware and LLM-based metrics, and analyze their sensitivity under controlled perturbations. Third, we plan to extend the study to additional AAS submodels and broader industrial domains to assess the generalizability of our findings. Finally, we aim to integrate the proposed metrics into adaptive evaluation pipelines that support continuous quality monitoring and improvement of AI-assisted AAS generation systems.

Acknowledgments. This work was supported by the research training group “Dependable AI Assistants for the Management of Dynamic Production Systems and Supply Chains (VAMoS)” at Mainz University of Applied Sciences and the Rheinland-Palatinate Technical University of Kaiserslautern-Landau, funded by the Ministry of Science and Health of Rhineland-Palatinate. The authors gratefully acknowledge this support.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Angelopoulos, A.N., Bates, S., Fannjiang, C., Jordan, M.I., Zrnic, T.: Prediction-powered inference. *Science* **382**(6671), 669–674 (2023). <https://doi.org/10.1126/science.adi6000>
2. Deng, S., Ma, Y., Zhang, N., Cao, Y., Hooi, B.: Information extraction in low-resource scenarios: Survey and perspective. In: 2024 IEEE International Conference on Knowledge Graph (ICKG). pp. 33–49 (2024). <https://doi.org/10.1109/ickg63256.2024.00013>
3. Es, S., James, J., Espinosa Anke, L.E., Schockaert, S.: RAGAs: Automated evaluation of retrieval augmented generation. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. pp. 150–158. Association for Computational Linguistics (Mar 2024). <https://doi.org/10.18653/v1/2024.eacl-demo.16>
4. Han, R., Peng, T., Yang, C., Wang, B., Liu, L., Wan, X.: An empirical study on information extraction using large language models. arXiv preprint arXiv:2305.14450 (2023). <https://doi.org/10.48550/arXiv.2305.14450>
5. Industrial Digital Twin Association: Specification of the asset administration shell part 1: Metamodel. Tech. Rep. IDTA 01001, IDTA (2023)
6. Industrial Digital Twin Association: Specification of the asset administration shell part 3a: Data specification – iec 61360. Tech. Rep. IDTA 01003-a-3-0, IDTA (2023)
7. Industrial Digital Twin Association: Aasx samples (nd), <https://admin-shell-io.com/samples/>, accessed: 2026-04-28
8. International Organization for Standardization: Iso/iec 25010:2011 systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models. Tech. rep., ISO (2011)
9. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020). <https://doi.org/10.48550/arXiv.2001.08361>
10. Kaya, F., Şanlı, E., Albayrak, Ö., Ünal, P., Kirci, P.: Asset administration shell tool comparison: A case study with real digital twins used in petrochemical industry. *Sensors* **25**(7) (2025). <https://doi.org/10.3390/s25071978>

11. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* **34**(1), 50–70 (2022). <https://doi.org/10.1109/TKDE.2020.2981314>
12. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out*. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), <https://aclanthology.org/W04-1013/>
13. Miller, G.A.: Wordnet: A lexical database for english. *Communications of the ACM* **38**(11), 39–41 (1995). <https://doi.org/10.1145/219717.219748>
14. Moradi, M., Samwald, M.: Evaluating the robustness of neural language models to input perturbations. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. pp. 1558–1570. Association for Computational Linguistics (Nov 2021). <https://doi.org/10.18653/v1/2021.emnlp-main.117>
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002). <https://doi.org/10.3115/1073083.1073135>, <https://aclanthology.org/P02-1040/>
16. Renze, M.: The effect of sampling temperature on problem solving in large language models. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. pp. 7346–7356. Association for Computational Linguistics (Nov 2024). <https://doi.org/10.18653/v1/2024.findings-emnlp.432>
17. Saad-Falcon, J., Khattab, O., Potts, C., Zaharia, M.: ARES: An automated evaluation framework for retrieval-augmented generation systems. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. pp. 338–354. Association for Computational Linguistics (Jun 2024). <https://doi.org/10.18653/v1/2024.naacl-long.20>, <https://aclanthology.org/2024.naacl-long.20/>
18. Sander, D.P., Dietz, L.: Exam: How to evaluate retrieve-and-generate systems for users who do not (yet) know what they want. In: *DESIRES*. pp. 136–146 (2021), <https://ceur-ws.org/Vol-2950/paper-16.pdf>
19. Vogel, J.A., Barth, C., Bayha, A., Braunisch, N., Garmaev, I., Ristin, M., Grüner, S.: Extraction of technical data using llms: Experience report and evaluation based on asset administration shells. In: *Automation 2025: Human-centric Automation*. pp. 153–168. VDI (2025)
20. Wagner, S., Goeb, A., Heinemann, L., Kläs, M., Lampasona, C., Lochmann, K., Mayr, A., Plösch, R., Seidl, A., Streit, J., Trendowicz, A.: Operationalised product quality models and assessment: The quamoco approach. *Information and Software Technology* **62**, 101–123 (2015). <https://doi.org/10.1016/j.infsof.2015.02.009>
21. Witte, C., Cimiano, P.: Intra-template entity compatibility based slot-filling for clinical trial information extraction. In: *Proceedings of the 21st Workshop on Biomedical Language Processing*. pp. 178–192. Association for Computational Linguistics, Dublin, Ireland (2022). <https://doi.org/10.18653/v1/2022.bionlp-1.18>
22. Xia, Y., Xiao, Z., Jazdi, N., Weyrich, M.: Generation of asset administration shell with large language model agents: Toward semantic interoperability in digital twins in the context of industry 4.0. *IEEE Access* **12**, 84863–84877 (2024). <https://doi.org/10.1109/ACCESS.2024.3415470>
23. Zhao, X., Deng, Y., Yang, M., Wang, L., Zhang, R., Cheng, H., Lam, W., Shen, Y., Xu, R.: A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys* **56**(11) (Jul 2024). <https://doi.org/10.1145/3674501>