

Performance evaluation of Large Language Models in extracting requirements sets from technical documents

Iris Graessler¹, Jan Pfeifer^{1*}

¹Heinz Nixdorf Institute, Chair for Product Creation, 33102 Paderborn

*Corresponding author. E-Mail address: jan.pfeifer@hni.upb.de; Tel.:05251/60 6265

Abstract: Manufacturers operate under dynamic regulatory conditions, leading to frequently changing requirements that must be managed with high effort. AI-based requirements extraction using Large Language Models (LLMs) has the potential to support engineers in this task by analyzing large documents, for example. Existing evaluation approaches in AI-supported requirements engineering mainly assess individual requirements and their consistency with source documents, while quality criteria of requirements sets are largely neglected. This paper presents a five-step approach to evaluate the performance of current LLMs in generating requirements sets: a systematic literature review to identify quality criteria of requirements sets (1), exemplary generation of requirements sets from the Machinery Directive 2006/42/EC using LLMs (2), examination (3), evaluation (4) and comparison (5) of the generated requirements sets. The results support engineers in selecting LLMs for creating requirement sets and in evaluating future LLMs.

Keywords:

Requirements Engineering, Artificial Intelligence, Large Language Models, Generative AI

1 Introduction

The process of engineering compliant technical systems is subject to complexity [1]. This complexity results, among other things, from numerous constraints that have to be considered throughout the entire Product Lifecycle (PLC) [2]. Technical documents, such as standards or regulations, are one possible form of documentation for these constraints. Technical documents are often extensive in content, written in natural language, complex and difficult to understand. Manually extracting requirements involves a considerable amount of effort. To provide engineers with validated knowledge in a machine-readable and semi-structured, rational and objective form, knowledge can be extracted automatically from technical documents. Using Artificial Intelligence (AI) promises great potential in requirements engineering.

Preliminary work has primarily addressed the technical implementation of AI-supported document analysis for extracting requirements. The BIKINI project, funded by the Federal Ministry of Research, Technology and Space of Germany, examined technical challenges in detail in using AI in requirements extraction. For example, GRAESSLER et al. [3] have identified challenges in AI-based requirement extraction. These challenges include, for example, different data types of source documents, different languages within a document, or quality of the training data for AI algorithms. These challenges have been overcome through a combination of approaches, applying new methods and technical advances in machine learning (ML). For example, augmented datasets can be used to improve quality of training data [4]. In further research by GRAESSLER et al. [5] and LUTTMER et al. [6], existing AI models for language processing, such as Bidirectional Encoder Representations from Transformers (BERT), are compared in terms of their performance in requirement extraction.

Large Language Models (LLMs) are one type of AI that is already being used in several applications in everyday engineering work. LLMs are not sufficiently considered in existing areas of research in requirements engineering. Many technical challenges mainly identified for generating individual requirements, such as document language, file type or formatting, have become secondary due to the rapid technical development of artificial intelligence. Applicability of AI can be shifted to larger contexts, such as entire sets of requirements, thanks to technical progress induced by LLMs. This results in a need for research, particularly in LLM-supported generation of high-quality sets of requirements rather than individual requirements. Two research questions (RQ) result from this clarification:

RQ1: What quality criteria exist for requirements sets?

RQ2: What level of performance do current LLMs achieve in generating requirements sets?

The paper at hand is structured as follows: First, the methodology used to answer the RQs is described. Based on a literature analysis, quality criteria of requirements sets are derived. Using a developed prompt, requirements sets are generated through three common LLMs. Generated requirement sets are then evaluated and compared in terms of fulfilling previously determined criteria of requirement sets.

2 Methodology

To answer the RQs, a five-step scientific methodology is applied. Following the DRM methodology [7], a prescriptive approach is defined: Under the assumption, that LLMs help to solve engineering tasks, the methodology includes generating requirement sets using LLMs and then comparing them with regard to data quality criteria. Based on existing approaches for engineering of requirements sets, a systematic literature review in accordance with PRISMA protocol [8] is conducted to identify quality criteria of requirements sets (1). Subsequently, market-leading LLMs are instructed to generate requirements sets based on the Machinery Directive 2006/42/EC [5] and considering identified criteria of requirements sets (2). The generated requirements sets are then analyzed in terms of their fulfilment of requirements set data quality criteria (3), individually evaluated according to the degree of criteria fulfilment (4) and compared (5).

3 Theoretical Foundation

Requirements management is the structured process of eliciting, documenting, analyzing, and managing requirements throughout the engineering of a system [9,10]. Requirements Engineering starts with an internal or external engineering order and focuses on identifying stakeholder needs, expectations, and constraints. These needs are translated into customer requirements and further refined into system requirements that define the conditions and capabilities a system or subsystem must fulfill [9]. Requirements management is important because it provides a clear and agreed-upon basis for system architecture, ensures compliance with standards and specifications, and supports controlled engineering decisions. Well-managed requirements reduce risks, misunderstandings, and costly changes later in the engineering process [10]. Regulatory documents are one source of requirements. Depending on industry, compliance with such regulatory requirements is essential for product approvals [10].

LLMs are large neural language models that are trained on billions of words [11]. Their size and use of transformer architecture enable them to analyze and generate language and solve a variety of tasks without task-specific training [12]. Through textual instructions known as prompts, LLMs can be instructed to perform specific tasks. Reference information can be added to the query in the form of document datasets [12]. Due to the complexity and increasing computing power of LLMs, previous machine learning benchmarks are no longer sufficient [13]. Modern tests for LLMs such as LiveBench [13] challenge LLMs with 18 tasks from the categories of math, coding, reasoning, language

comprehension, instruction following, and data analysis. General LLM tests such as LiveBench provide an overview of a model's performance, but they do not replace application-specific tests.

4 Generation of Requirements Sets

In the following, requirements sets generated by common LLMs which are market-leading in industry according to KPMG [14] are compared in terms of their resulting data quality. The three LLMs are OpenAI GPT5.0, Meta LLaMA 3.3 70B, and Google Gemma 3 27B (Gemini). First, approaches in literature are identified that enable to evaluate the quality of requirements, typically proposing a set of evaluation criteria. These approaches are then analyzed for specific criteria of requirements sets. As a result regarding RQ1, a set of criteria of requirements sets generated by LLMs is created. A prompt is developed considering identified quality criteria of requirements sets. The prompt instructs the LLMs to generate sets of requirements based on a requirements stereotype developed in workshops. The generated requirements sets are then evaluated and compared.

4.1 Identification of relevant approaches

To determine quality criteria of requirements sets, two topics are covered in literature review according to PRISMA [8]. *Requirements engineering* and *evaluation schemes* are combined through search vectors into a search string. Literature identified in Scopus and Web of Science is filtered according to the relevance specifically targeting requirements sets, evaluability of the requirements set criteria and relevance to mechanical engineering. For example, literature is filtered out that deals primarily with quality of requirements and not with quality of requirements sets. Based on title and abstract, the initial total of 535 literature references includes 21 relevant references, and after detailed review a final number of nine references.

In product engineering, comprehensive requirement criteria are defined in norms and standards to ensure quality of individual requirements and requirements sets. All nine identified references refer to norms and standards. For example, ISO/IEC/IEEE 29148 [15] distinguishes between criteria of individual requirements, requirements sets and requirement language. Criteria of requirements sets include *complete, consistent, feasible, comprehensible and able to be validated*. Criteria of requirements language include *avoiding superlatives, subjective language, vague pronouns, ambiguous terms such as adverbs and adjectives, open-ended terms, comparative phrases, loopholes and terms that imply totality*. Other basic literature besides norms and regulations such as the INCOSE handbook [2] refer to corresponding standards and norms to describe requirement criteria. In industry-oriented, application-related literature such as [1], a reduced set of quality criteria are used to ensure pragmatic applicability. The criteria are *complete, clear, consistent and correct*.

In software engineering, there are more comprehensive quality criteria of requirements and requirements sets. For example, BOKHARI and SIDDIQUI [16] summarize Software Requirements Specification as a key document describing external behavior of a software system and its role in achieving successful software engineering. The study consolidates essential quality criteria of requirements documents, distinguishing internal attributes – e.g., *unambiguity, correctness, completeness, consistency, verifiability, modifiability, traceability, precision, and appropriate level of detail* – from external attributes – e.g., *achievability, conciseness, design independence, reusability, and electronic storage*. Together, these criteria provide a structured framework for producing high-quality, maintainable, and effective requirements specifications. HALLIGAN [17] defines and systematizes requirements quality through a set of ten requirements quality factors that collectively characterize well-formed requirements. The factors *correctness, completeness, consistency, clarity, non-ambiguity, connectivity, singularity, testability, modifiability, and feasibility* capture both semantic and structural properties necessary for requirements to be understandable, implementable, and verifiable. HALLIGAN further introduces a

structural template for natural-language requirements, decomposing each statement into standardized elements such as actor, conditions, action, object, and constraints [17]. With help of criteria, sets of requirements generated by LLM can be systematically evaluated for quality.

4.2 Analyzing relevant approaches regarding requirements sets

Based on the identified nine approaches, a set of criteria of requirements is compiled answering RQ1. The set of criteria includes criteria that relate to entire requirements sets. Criteria of individual requirements like linguistic criteria are not included. As a review result of relevant literature, the subsequent criteria of requirements are determined: *complete*, *consistent*, *feasible*, *clear*, *testable* and *correct*. These criteria are laid down in standards (e.g. [18]) and are frequently cited in the nine identified publications on requirements engineering. A list of all criteria considered including description and reference is shown in Table 1.

Table 1: Quality criteria of requirements sets

Criteria	Description	Reference
complete	The requirements set needs no further amplification because it contains everything pertinent to the definition of the system or system element being specified. In addition, the set contains no To Be Defined (TBD), To Be Specified (TBS), or To Be Resolved (TBR) clauses. Resolution of the TBx designations may be iterative and there is an acceptable timeframe for TBx items, determined by risks and dependencies.	[1,2,18]
consistent	The requirements set does not have individual requirements which are contradictory. Requirements are not duplicated. The same term is used for the same item in all requirements.	[1,2,18]
feasible	The requirements set can be satisfied by a solution that is obtainable/affordable within life cycle constraints (e.g., cost, schedule, technical, legal, regulatory).	[2,18]
clear	The requirements set is stated in such a way so that it can be interpreted in only one way. The requirements set is stated simply and is easy to understand.	[2,18]
testable	The requirements set has the means to prove that the system satisfies the specified requirements set. Evidence may be collected that proves that the system can satisfy the specified requirements set. Verifiability is enhanced when the requirements set is measurable.	[2,18]
correct	The requirements set is upwards traceable to specific documented stakeholder statement(s) of need, higher tier requirements set, or other source (e.g., a trade or design study).	[1,2,18]

4.3 Criteria for structuring requirements sets

To generate comparable requirements sets, an output format must first be specified. The format of the requirements set is based on DIN EN ISO/IEC/IEEE 29148 [18] and was modified in workshops with industry partners for the [anonymous] project founded by the European Union. As part of the project, a knowledge base is to be established that will enable requirements to be integrated into engineering processes according to their validity. The table format is illustrated in Table 2.

Table 2: Requirements output format

ID	Description	Validation criterion	Life Cycle Phase	Time of application	Geo. Area of application	Reference to source	Comment

Each requirement is identified in the first column by an ID composed of the name of the technical document and a corresponding number. In the second column, the requirement is described in accordance with the SOPHIST template [19], providing a clear and structured formulation. A validation

criterion is defined in the third column, outlining a measurable property a product must fulfill to satisfy the requirement. Assignment to a particular phase of the product life cycle according to the generic Product Lifecycle (gPLC) [20] is described in the fourth column, which covers strategic planning, engineering, realization, operation/ service delivery, and decommissioning. The fifth column specifies the applicable time frame or period of validity for the requirement. The sixth column describes where the requirement applies. Reference information is provided in the seventh column, identifying the source technical document and the exact location within that document from which the requirement was extracted. Additional comments and other relevant details, such as cross-references to technical documents, are included in the eighth column.

4.4 Development of a prompt for the generation of requirements sets

A comprehensive prompt is an essential part for the extraction of requirements with LLMs [12]. The prompt was developed, iterated and optimized considering guidelines for developing prompts, for example from Google [21] or OpenAI [22]. Iterations consisted, for example, of adapting context examples or adding special instructions that activate deep thinking processes. The prompt is presented and explained below. All components are described sequentially.

The **first block** describes the basic role that the LLM is to assume. In case of requirements extraction, this is the role of a requirements engineer:

“You are a requirements engineer and are tasked with extracting requirements set from the following document.”

The **second section** describes the procedure. Two basic quality criteria are specified: Firstly, requirements should relate to a specific technical system. A positive and a negative example are given. Secondly, requirements should be extracted in full. The list of requirements should neither be shortened nor idealized.

“Analyze the document and identify all requirements that relate to a specific system to be engineered. Good example: The system to be developed must contain 50% recycled material. Bad example: The development team must consist of 5 people. The most important criterion is that you really extract all requirements. No abbreviated or idealized list.”

The **third section** describes the structure of the eight-column results table. Specific examples are given for each row. The exemplary instruction for the first column is as follows:

“The requirements should be displayed in a table with the following columns: Column 1: Requirement ID with reference to the content of the document. Do not simply refer to the data name but refer to the content. Good: ISO14001-001 Bad: ISO14001-001.pdf.”

The **fourth section** describes specific quality criteria (Table 1) for the requirements set. They are listed with relevant examples. For example:

“The requirements, both collectively and individually, should meet the following criteria: Completeness: No requirement should be missing. All requirements should be extracted from all sections of the document. Also list requirements that refer to other documents.”

The **fifth block** describes the concrete formulation of requirements. All requirements should be formulated according to the SOPHIST template:

“Formulate the requirements in the SOPHIST template: The [subject] must/should/may [predicate] [object] condition/qualification.”

In the **sixth block**, the allocation of life cycle phases is specified. The individual phases are presented and explained with the help of short examples. For example:

“Assign the appropriate lifecycle phase to each requirement: Strategic planning, engineering, realization, use, decommissioning. ‘Strategic planning’ refers to all content that must take place before the actual development, e.g. business orientation.”

At the end of the prompt, general conditions are listed. For example, tables should be exportable. One example is further processing in Microsoft Excel, an alternative could be XML:

“Please provide me with the table in a format that I can copy into Excel.”

4.5 Examination, evaluation and comparison of the extracted requirements sets

In the following, the requirements sets extracted by the LLM models are analyzed, evaluated, and then compared. Examples of requirements from the generated requirements sets with the described prompt are first presented. Requirements sets are then evaluated according to the criteria listed in Table 1.

In Table 3, examples of requirements from the requirements set created by ChatGPT are shown.

Table 3: Excerpt from the ChatGPT requirements set (4 of 26 requirements)

Requirement ID	Requirement Text (SOPHIST Template)	Validation Criterion	Lifecycle Phase	Temporal Validity	Geographical Validity	Source Reference
MD2006-AnnI-1.1.1-001	The system must be designed and constructed so that it can be operated, adjusted, and maintained without putting persons at risk under the intended conditions of use.	No hazardous situation identified during operation, adjustment, or maintenance per risk assessment.	Engineering	From 2009-12-29	EU / EEA	Annex I, 1.1.1 —
MD2006-AnnI-1.1.2-001	The system must be designed and constructed taking into account the principles of safety integration.	Risk reduction hierarchy demonstrably applied (inherently safe → safeguards → information).	Engineering	From 2009-12-29	EU / EEA	Annex I, 1.1.2 —
MD2006-AnnI-1.1.2-002	The system must eliminate or reduce risks as far as possible by design and construction.	Residual risk list minimized and justified.	Engineering	From 2009-12-29	EU / EEA	Annex I, 1.1.2 —
MD2006-AnnI-1.1.2-003	The system must incorporate necessary protective measures against risks that cannot be eliminated by design.	Protective measures verified against all residual risks.	Engineering	From 2009-12-29	EU / EEA	Annex I, 1.1.2 —

The **ChatGPT** requirements set represents a comprehensive and systematic transformation of the relevant provisions of Directive 2006/42/EC into system-level, verifiable requirements for a technical system. In the evaluation, the extracted requirements are first compared with the original specifications in the regulations. Ground truth is defined by manual extraction of requirements from the regulation, and verification by three experts from industry. The requirements set is *complete*. The requirements set describes required capabilities, characteristics, constraints, and safety-related quality factors necessary to achieve compliance with the essential health and safety requirements from Directive 2006/42/EC, and it contains no TBD, TBS, or TBR placeholders. While the set does not aim to define application-specific functional or performance characteristics such as payload, reach, accuracy, or cycle time, this limitation is *consistent* with the nature of the source document and does not reduce the completeness of the

compliance baseline. The SOPHIST template was only adhered partially. While the basic scheme was followed, in some cases (e.g. MD2006-AnnI-1.1.1-001) several individual requirements were mixed into one aggregated requirement. Requirements are internally consistent and structured. Each requirement is uniquely defined, does not conflict with or duplicate other requirements, and uses homogeneous terminology, measurement concepts, and verification logic that align closely with the language and intent of the Machinery Directive. Potential overlaps inherent in the source text have been resolved through atomic decomposition, resulting in a coherent and non-contradictory set. From a *feasibility* perspective, the requirements seem realistic and achievable within typical technical, cost, and schedule constraints for technical systems. The final feasibility depends on the specific system and company and cannot be assessed without taking into account the context, which has not yet been considered. The requirements set is also *clear* and understandable, both in terms of individual requirement statements and their relation to the overall system and its lifecycle. Explicit lifecycle phase assignment, and inclusion of validation criteria make expectations transparent and support traceability and engineering decision-making. Furthermore, the requirements are *testable*. Each requirement can be verified through inspection, analysis, testing, or documentation review, enabling systematic verification and validation activities. However, specific validation criteria must be determined for each individual application. Finally, the set is *correct*. All requirements are directly derived from the Machinery Directive and accurately reflect its normative “shall” and “must” statements without introducing additional or speculative content. Overall, the requirements set constitutes a high-quality, regulation-driven system requirements baseline for technical systems.

In Table 4, examples of requirements from the requirements set created by Gemini are shown.

Table 4: Excerpt from the Gemini requirements set (4 of 21 requirements)

Requirement ID	Requirement Text (SOPHIST Template)	Validation Criterion	Lifecycle Phase	Temporal Validity	Geographical Validity	Reference in Source
MD2006/42-EHSR-1.1.2	The system must be designed to be fit for its function and operated without risk under intended conditions.	Safety factor ≥ 1.1 during nominal load tests	Engineering	From 29 Dec 2009	EU / EEA	Annex I, 1.1.2 (a)
MD2006/42-EHSR-1.1.3	The system must consist of materials that do not endanger the health or safety of persons (e.g., toxic dust).	REACH/RoHS compliance certificates	Engineering	From 29 Dec 2009	EU / EEA	Annex I, 1.1.3
MD2006/42-EHSR-1.1.5	The system must be designed to be handled safely (e.g., lifting points for installation).	Presence of clearly marked lifting eye bolts	Engineering	From 29 Dec 2009	EU / EEA	Annex I, 1.1.5
MD2006/42-EHSR-1.2.1	The system must feature control systems designed to prevent hazardous situations from occurring.	Performance Level (PL) r achieved according to ISO 13849-1	Engineering	From 29 Dec 2009	EU / EEA	Annex I, 1.2.1

The **Gemini** requirements set forms a comprehensive and well-structured compliance-oriented system requirements baseline derived from Directive 2006/42/EC. The set is nearly *complete*. Some requirements are missing like, for example, that the system must be equipped with an emergency stop. The set contains no undefined or open placeholders such as TBD, TBS, or TBR. The requirements are internally *consistent* and logically coherent. Each requirement is uniquely identified, unambiguous, and free of conflicts or contradictions. Cross-references are used appropriately and indicate conceptual

dependencies without overlapping. Terminology, structure, and units are used consistently throughout the set and align with both the Machinery Directive and established engineering and safety standards. From a *feasibility* standpoint, all requirements are realistically achievable. Specified validation criteria such as load testing, Performance Level achievement, electrical safety testing, noise measurement and documentation checks can be implemented within acceptable cost, schedule, and technical risk constraints. The requirements are *clearly* formulated using the SOPHIST template, which makes expectations explicit and reduces interpretational ambiguity. In addition, the requirements are fully *testable*. Each requirement can be verified through inspection, analysis, testing, or demonstration. Finally, the requirements set is *correct*. Every requirement is directly traceable to a specific clause of Directive 2006/42/EC and accurately reflects its normative intent without introducing additional or speculative content. In Table 5, examples of requirements from the requirements set created by LLaMA are shown.

Table 5: Excerpt from the LLaMA requirements set (4 of 9 requirements)

Requirement ID	Requirement Text (SOPHIST Template)	Validation Criterion	Lifecycle Phase	Temporal Validity	Geographical Validity	Reference in Original Document
MD-001	The manufacturer must operate an approved quality system for design, manufacture, final inspection and testing	Approved quality system	Realization	from 2006	all EU countrys	ANNEX X
MD-002	The technical file must demonstrate that the machinery complies with the requirements of this Directive	Compliance with Directive	Engineering	from 2006	all EU countrys	ANNEX VII, Part A
MD-003	The manufacturer must take all measures necessary to ensure compliance of the manufactured machinery with the technical file and the requirements of this Directive	Compliance with technical file and Directive	Realization	from 2006	all EU countrys	ANNEX VIII
MD-004	The quality system must ensure conformity of the machinery with the provisions of this Directive	Conformity with Directive	Engineering	from 2006	all EU countrys	ANNEX X

The **LLaMA** requirements set shows significant deficiencies. In terms of *completeness*, the set is clearly insufficient. A substantial number of essential requirements are missing, and large parts of the source document were not analyzed or translated into system requirements. In addition, a technical system is not always the reference object of the requirements set. In some cases, requirements are placed on the manufacturer rather than on the system to be engineered (e.g. MD-003). *Consistency* is also limited. While no direct contradictions are evident, the lack of systematic coverage and the incomplete analysis of the source material lead to gaps and conceptual discontinuities within the set. *Feasibility* cannot be reliably assessed, as many requirements are formulated at a high level of abstraction without sufficient technical detail. The *clarity* of the requirements set is reduced by vague wording and absence of a structured template comparable to SOPHIST template. As a result, expectations are not always explicit, leaving room for interpretation. *Testability* of the requirements set is weak. Validation criteria are often imprecise or missing altogether, making objective verification through inspection, analysis, or testing difficult or impossible. Finally, correctness is not ensured. Due to the incomplete consideration of the

original directive and missing traceability to specific clauses, it is unclear whether the requirements accurately reflect the normative intent of the source document.

In the following (Figure 1), the performance of LLMs is compared based on their fulfillment of criteria of requirements sets, based on a three-point scale, with a value of 0 representing poor fulfillment and a value of 2 representing good fulfillment of the criterion.

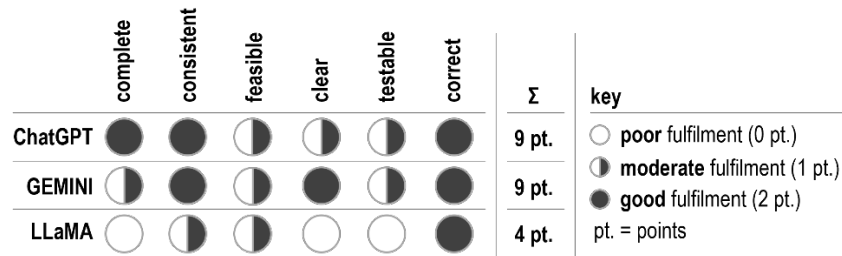


Figure 1: Comparison of LLM performance

Comparing ChatGPT, Gemini, and LLaMA requirements sets reveals clear differences in quality across all key evaluation criteria: *completeness*, *consistency*, *feasibility*, *clarity*, *testability*, and *correctness*. The **ChatGPT** set achieves a “good” rating in three out of six criteria. The ChatGPT requirements set provides a comprehensive, systematically structured, and fully traceable set of requirements for a system, with clear, testable, and feasible statements. Usage of the SOPHIST template and generic validation criteria can be improved. **Gemini** also delivers a strong performance, with most criteria rated “good” except for completeness, which was rated slightly lower as “Good” due to a few missing requirements, such as the emergency stop. The set is internally consistent, feasible, clear, testable, and correct, with well-defined verification methods and traceability to the directive. In contrast, the **LLaMA** requirements set exhibits significant deficiencies across all criteria. Completeness and correctness are rated “poor”, as many essential requirements are missing, the source document is only partially analyzed, and traceability to the directive is lacking. Consistency, feasibility, and clarity are also rated low with gaps and vague formulations that hinder practical implementation. Testability is severely limited, as validation criteria are imprecise or absent, preventing reliable verification. Overall, LLaMA falls far short of standards demonstrated by ChatGPT and Gemini, making it unsuitable as a compliance-oriented system requirements baseline.

This comparison highlights that while ChatGPT and Gemini provide structured, high-quality requirements sets suitable for a baseline for regulatory compliance, LLaMA’s approach leads to incomplete, ambiguous, and largely untestable requirements, emphasizing the critical importance of systematic methodology and traceability in requirement generation. Regarding RQ2, in results show that the LLMs ChatGPT and Gemini can generate usable requirements sets. However, these are not flawless and must be reviewed and adjusted by humans. Research also has determined that the criteria “feasibility” and “testability” depend on the corresponding engineering use case. These must be added to future prompts to further improve quality of generated requirements sets. One approach, for example, would be to provide the LLM with project-specific data such as parts lists. Furthermore, quality of extracted requirements sets from other technical documents must be evaluated.

5 Conclusion and outlook

Increasing complexity of technical systems requires systematic extraction of requirements from technical documents, which are often lengthy, complex, and difficult to understand. Manual extraction is time-consuming, making AI, particularly LLMs, a promising solution. The study follows a five-step scientific methodology: first, quality criteria of requirements sets are identified through a systematic

literature review; then, leading LLMs are instructed to generate requirements from the Machinery Directive 2006/42/EC. Generated requirements sets are subsequently analyzed, evaluated, and compared. Based on the literature, six criteria of requirements sets were defined: completeness, consistency, feasibility, clarity, testability, and correctness. An iteratively developed prompt ensures that LLMs systematically capture all relevant requirements. Three LLMs were evaluated: ChatGPT, Gemini, and LLaMA. ChatGPT and Gemini generated comparatively good requirements sets with deficits in feasibility and testability, while LLaMA showed significant deficiencies, including missing, ambiguous, and untestable requirements. These findings highlight the importance of a systematic methodology, clear quality criteria, and traceability for automated requirement generation. Results confirm usefulness and applicability of LLMs in requirements engineering. The evaluation procedure can be transferred or extended to future LLMs. The next step is to investigate how the quality of LLM-generated requirement sets can be further improved. To this end, new models and model types will be examined, as well as the inclusion of retrieval-augmented generation systems. This improvement can enhance the quality of requirement sets in terms of feasibility and testability.

References

- [1] I. Gräßler and C. Oleff, *Systems Engineering Berlin, Heidelberg*: Springer Berlin Heidelberg, 2022.
- [2] INCOSE, *INCOSE Systems Engineering Handbook* Wiley, 2023.
- [3] I. Graessler, D. Preuss and J. Pottebaum, “Extrahierung von Anforderungen aus natürlich-sprachlichen Lastenheften: Was erschwert eine KI-basierte Extrahierung?,” *Softwaretechnik-Trends* Band 42, Heft 1., 2022.
- [4] I. Graessler, D. Preus, L. Brandt and M. Mohr, “Efficient Extraction of Technical Requirements Applying Data Augmentation,” in *2022 IEEE International Symposium on Systems Engineering (ISSE)*: IEEE, 2022, pp. 1–8.
- [5] I. Graessler, D. Oezcan and D. Preuss, “AI-based extraction of requirements from regulations for automotive engineering,” in *DS 125: Proceedings of the 34th Symposium Design for X (DFX2023)*: The Design Society, 14 and 2023, pp. 163–172.
- [6] J. Luttmer, V. Prihodko, D. Ehring and A. Nagarajah, “Requirements extraction from engineering standards – systematic evaluation of extraction techniques,” *Procedia CIRP*, vol. 119, pp. 794-799, 2023.
- [7] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, ..., “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews,” *BMJ (Clinical research ed.)*, vol. 372, n71, 2021.
- [8] VDI/VDE, “Development of mechatronic and cyber-physical systems. VDI/VDE 2206:2021,” 2021.
- [9] C. Rupp, *Requirements-Engineering und -Management. Das Handbuch für Anforderungen in jeder Situation*, 7., aktualisierte und erweiterte Auflage München: Hanser, 2021.
- [10] A. Tamkin, M. Brundage, J. Clark and D. Ganguli, “Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models,” 2021.
- [11] OpenAI, “GPT-4 Technical Report,” 2023.
- [12] C. White, S. Dooley, M. Roberts, ..., “LiveBench: A challenging contamination-limited LLM benchmark,” 2025.
- [13] KPMG International Limited, “Woher bezieht ihr Unternehmen Large Language Models?,” 2024.
- [14] 2023 IEEE 31st International Requirements Engineering Conference (RE) (IEEE, 2023).
- [15] M. U. Bokhari and S. T. Siddiqui, “Metrics for Requirements Engineering and Automated Requirements Tools,” 2011.
- [16] R. J. Halligan, “Requirements Quality Metrics: The Basis of Informed Requirements Engineering Management,” 1993.
- [17] IEEE, *Systems and software engineering — Life cycle processes — Requirements engineering (29148)* .
- [18] SOPHISTen and R. Kluge, “MASTER: Schablonen für alle Fälle,” 2024.
- [19] I. Gräßler and J. Pottebaum, “Generic Product Lifecycle Model: A Holistic and Adaptable Approach for Multi-Disciplinary Product–Service Systems,” *Applied Sciences*, vol. 11, no. 10, p. 4516, 2021.
- [20] Google, “Prompting guide 101. A quick-start handbook for effective prompts,”
- [21] OpenAI, “Best practices for prompt engineering with the OpenAI API,” [Online]. Available: <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api>.