Evaluating the Impact of Input Text Properties on LLM-Based Requirements Extraction

Konstantin Valeev[1,2], Fabiano Dalpiaz[1], Fatma Başak Aydemir[1]
[1]Utrecht University, Utrecht, The Netherlands
[2]JetBrains, Amsterdam, The Netherlands

***Abstract*—Large Language Models (LLMs) are increasingly applied to requirements engineering tasks, yet existing benchmarks measure model performance without accounting for how properties of the input text affect extraction outcomes. Among the broad spectrum of requirements engineering activities, this research focuses on *requirements extraction*, identifying requirement statements in existing documents, a task that can be evaluated against human-annotated ground truth. We argue that measurable text properties can indicate extraction performance. We propose a three-stage research program. Stage 1 defines a metrics toolkit spanning five dimensions (entity coherence, readability, structure, requirements readiness, domain terminology) and validates their discriminative power on a corpus of 376 documents: 19 metrics were used for statistical analysis; all 19 significantly distinguish document types (Welch's ANOVA with Benjamini–Hochberg FDR correction, $p < 0.05$), achieving 93.09% classification accuracy. Stage 2 will assemble a dataset of natural-language documents paired with human-annotated requirement statements, then measure how each document's metric profile correlates with LLM extraction quality. Stage 3 will compare multiple LLMs and prompting strategies to map technique-specific failure boundaries. The envisioned outcome is a decision framework mapping document profiles to expected extraction quality. We report Stage 1 results and the research roadmap.**

***Index Terms*—requirements engineering, requirements extraction, large language models, text difficulty, metrics, natural language processing**

I. Introduction

Consider two documents. The first is a software requirements specification (SRS): "The system shall authenticate users via two-factor authentication within 5 seconds." The second is an excerpt from a product brainstorming session: "We really need to make sure people can log in fast, maybe with their phone or something, because users keep complaining about how slow it is." Both documents contain a requirement about authentication. However, any requirements engineer would recognize that extracting the requirement from the first document is trivial (and could be framed as requirements demarcation [1]), while extracting a well-formed requirement from the second one demands interpretation, disambiguation, and reformulation.

Requirements engineering (RE) encompasses a broad set of activities: elicitation, analysis, specification, validation, and management, each presenting distinct challenges for automation. LLMs have been applied across this spectrum, from generating user stories [2] and automating requirements elicitation from interview transcripts [3] to detecting ambiguity in requirements [4]. This research focuses specifically on *requirements extraction*: the task of identifying and isolating individual requirement statements from existing natural-language documents. Extraction is a narrower task than elicitation, which involves discovering requirements through interaction with stakeholders (interviews, workshops, observation). In extraction, the requirements are assumed to already exist in written form, and the challenge is locating and formulating them from documents that vary widely in structure, formality, and clarity. This narrower scope makes the task feasible for systematic evaluation: the input is a document, the output is a set of requirement statements, and quality can be measured against human-annotated ground truth.

When an LLM fails to extract requirements from the second document mentioned above, the cause is ambiguous: the failure may reflect a weakness of the model, or an inherent property of the input text. Current evaluations of LLMs for RE tasks [5] tend to benchmark models on curated datasets without systematically characterizing input difficulty. There is a lack of methods that separate model capability from text complexity. This limits the ability of researchers and practitioners to select appropriate extraction techniques for specific document types and to understand the operational boundaries of current approaches.

This paper proposes treating text difficulty as an independent variable in the evaluation of LLM-based requirements extraction. The contributions are:

1) A toolkit that computes metrics across five dimensions, designed to characterize the potential difficulty of extracting requirements from a given document. After removing highly correlated readability metrics (when $r > 0.95$), 19 independent metrics are retained.
2) A preliminary feasibility validation on a 376-document corpus, demonstrating that the metrics capture meaningful variation across document types.
3) A three-stage research roadmap: from metric validation through single-LLM correlation to multi-LLM comparison, to map technique-specific failure boundaries.

The central hypothesis is: measurable text properties correlate with the quality of LLM-based requirements extraction, and different LLMs exhibit different failure boundaries along the text difficulty spectrum.

The research is structured in three stages. Stage 1 (reported here) defines and validates the metrics as a foundation. Stage 2 will correlate metrics with extraction quality for individual

LLMs. Stage 3 will compare multiple LLMs to identify where each technique degrades. The envisioned outcome is a practical mapping: given a document's metric profile, which extraction approach is most likely to succeed?

II. Related Work

NLP for RE. Natural language processing (NLP) has a long history in RE, from early rule-based approaches to modern deep learning methods. Surveys by Zhao et al. [5] and Ferrari et al. [6] cover requirements classification, requirements extraction, and requirements quality assessment. Recent work has shifted toward transformer-based models and LLMs for these tasks.

LLMs in RE. A recent systematic literature review of 74 LLM-for-RE studies [7] confirms that the field has rapidly adopted LLM-based approaches, predominantly using zero-shot and few-shot prompting, yet evaluations remain largely confined to controlled settings with limited attention to input characteristics. Several studies have applied LLMs to RE tasks including requirements classification, ambiguity detection, and requirements extraction [8], as well as traceability link recovery [9] and dependency detection [10]. However, most evaluations report aggregate accuracy on benchmark datasets without analyzing how properties of the input requirements artifacts affect performance. Even recent work that evaluates prompting strategies across multiple LLMs, such as TraceLLM's multi-model traceability evaluation [9], which found that performance depends critically on prompt design, does not control for measurable properties of the input artifacts. There is currently no established method for characterizing the difficulty of an input document prior to extraction. The present work addresses this gap.

Text Readability and Complexity. Readability formulas originated with Flesch [11] and were refined by Gunning [12], Kincaid et al. [13], Coleman and Liau, and McLaughlin [14], among others. These formulas are well-established for assessing general text complexity, but have not been systematically applied as predictors of NLP task difficulty in the RE domain. Prior applications of readability metrics to requirement artifacts have focused on quality assessment rather than on predicting automated extraction performance.

Entity Coherence. Barzilay and Lapata [15] introduced the Entity Grid model, which tracks how entities transition between grammatical roles (subject, object, other, absent) across consecutive sentences. The model has been applied to document quality assessment and text ordering tasks. The present work adopts the Entity Grid via the TRUNAJOD library [16] to measure coherence as one dimension of text difficulty for requirements extraction.

Requirements Quality Metrics. IEEE 29148 [17] defines characteristics of well-formed requirements, including the use of modal verbs ("shall", "must") and consistent subject-verb-object structure. The EARS (Easy Approach to Requirements Syntax) framework [18] provides templates for writing unambiguous requirements. The IREB CPRE glossary [19] standardizes RE terminology across 198 terms. Prior work has used these standards to assess requirements specification quality [20], but not as predictors of automated extraction difficulty.

Gap. To the best of our knowledge, no prior work systematically treats measurable text properties as independent variables for predicting LLM-based requirements extraction quality. Existing evaluations of LLMs for RE report model performance metrics without characterizing what makes some input requirements artifacts harder to process than others. The present work addresses this gap by proposing an initial framework of metrics across five dimensions and a research program to validate their predictive power.

III. Approach: Text Difficulty Metrics

A. Design Principles

The toolkit computes 25 raw metrics per document across five analyzers. After combining six highly inter-correlated readability formulas into a single consensus measure, 19 metrics are retained for statistical analysis.

The toolkit measures observable text properties such as sentence length, structural markers, vocabulary patterns—without attempting to assess semantic meaning or correctness. The decision to limit measurement to surface properties is intentional: the toolkit captures what can be measured automatically and reproducibly, leaving semantic assessment to future work. Full implementation details, including regular expression patterns, lexicon lists, and configuration options, are documented in the repository (see Data Availability Statement).

B. Five Dimensions

We identified five dimensions that capture the main aspects of text that we hypothesize affect an LLM's ability to locate and extract requirement statements. The selection was guided by three sources: (1) established RE quality frameworks: IEEE 29148 [17] defines well-formed requirements statement format, which informed the Requirements Readiness dimension, and usage of CPRE glossary [19] terms is considered to be a predictor of text written in the form of well-stated requirements; (2) the NLP and computational linguistics literature on text difficulty—readability formulas [11], [13] and entity coherence models [15] are widely used measures of text complexity that have not previously been applied as predictors of RE task performance; and (3) the practical observation that document format (headings, lists, markup) provides positional cues that LLMs may exploit during extraction, motivating the Structure dimension. Specifically, source (1) informed dimensions D4 (Requirements Readiness) and D5 (CPRE Glossary); source (2) motivated D1 (Entity Coherence) and D2 (Readability); source (3) motivated D3 (Structure). Together, these five dimensions span a gradient from low-level surface features (readability, structure) through discourse-level organization (entity coherence) to domain-specific RE patterns (requirements readiness, CPRE terminology usage).

D1. Entity Coherence (8 metrics). Based on the Entity Grid model [15], implemented via the TRUNAJOD library [16] and spaCy. TRUNAJOD was selected because it is, to the

best of our knowledge, the only publicly available Python implementation of the Entity Grid model; it computes transition probabilities via spaCy's `en_core_web_sm` dependency parser. The analyzer tracks how entities transition between grammatical roles (Subject, Object, Other, Absent) across consecutive sentences, computing four coherence scores plus entity density and persistence measures. Low coherence indicates that entities appear and disappear unpredictably, reducing the contextual information available to the LLM.

D2. Readability (1 retained metric, reduced from 6). Six established readability formulas—Flesch Reading Ease, Flesch–Kincaid, ARI, Gunning Fog, SMOG, Coleman–Liau—are computed internally. However, correlation analysis revealed that five of the six formulas are inter-correlated at $r > 0.95$ (e.g., Flesch–Kincaid vs. Gunning Fog $r = 0.986$). To avoid multicollinearity in the statistical analysis, only their average (`consensus_grade_level`) is retained as the single readability metric. Both extremes of the readability spectrum may hinder extraction: highly complex text is difficult to decompose into discrete requirements; overly simple text may lack the precision needed to express requirements unambiguously.

D3. Structure (4 metrics). Counts structural markers: headings and list items across multiple formats (Markdown, HTML, ALL CAPS, numbered sections). The analyzer auto-detects document format using format-specific regular expressions: Markdown patterns (e.g., `#+ , - , * `), HTML tags (`<h[1-6]>`, `<li>`), and plain-text conventions (ALL CAPS lines, numbered paragraphs). Structural markers provide positional cues about where requirements are located. Without headings and lists, the LLM must rely solely on content analysis to identify requirements in continuous prose.

D4. Requirements Readiness (5 metrics). Measures the degree to which the text uses formal requirement patterns as defined in IEEE 29148. The SMAO (Subject-Modal-Action-Object) detector identifies sentences matching the pattern "[Subject] [shall/must/will] [verb] [object]" using spaCy dependency parsing. Additional metrics capture sentence length consistency (`sentence_length_cv`), terminological ambiguity (39 vague terms such as "etc.", "user-friendly", "fast"), and anaphoric pronoun density (12 pronoun types). Of the five dimensions, requirements readiness is the most directly relevant to extraction: a high `smao_percentage` indicates that requirements are already expressed in standard form.

D5. CPRE Glossary (1 metric). Counts occurrences of 198 standardized RE terms from the IREB CPRE glossary, spanning 7 term categories (Requirements Types, Artifacts, Activities, Quality Criteria, Modeling, Roles, Concepts). Term matching uses case-insensitive regular expressions with word-boundary anchors to avoid partial matches. Documents that use standard RE vocabulary signal requirements more explicitly than documents using informal, domain-specific language.

IV. Preliminary Evaluation

The preliminary evaluation validates that the proposed metrics capture meaningful variation across document types. This constitutes a feasibility validation: if the metrics cannot distinguish between documents that intuitively differ in extraction difficulty (e.g., structured SRS vs. Wikipedia articles), they would be unlikely to predict LLM extraction performance. We emphasize that Stage 1 validates discriminative power among document categories, not prediction of extraction difficulty, which is the subject of Stage 2.

A. Corpus Construction

The toolkit was applied to a corpus of 376 documents in five categories, selected to span a range from requirement-focused artifacts to non-requirement text.

TABLE I
Stage 1 Corpus Composition

Category	Source	Files	Description
Short PRD	GitHub repos	225	Short PRDs, 1–3 pp.
PURE	PURE dataset [21]	76	Raw markdown
GitHub Articles	GitHub repos	51	Non-requirements
Wikipedia	Wikipedia	13	Software excerpts
Long PRD	GitHub repos	11	Full PRDs, 10+ pp.

The five categories were chosen to represent a gradient of expected extraction difficulty. Short PRD and Long PRD (Product Requirements Document) contain documents explicitly written as requirements artifacts, with structural markers (headings, numbered lists) and partially in requirement-style language. PURE contains SRS documents in raw markdown form. GitHub Articles and Wikipedia Articles serve as negative controls—documents that may discuss software systems but are not structured as requirements specifications. This gradient tests whether the metrics can separate documents along the dimension most relevant to the research: proximity to formal requirements writing.

The categories represent a sample with imbalanced sizes (225 vs. 11), which we acknowledge as a limitation. The imbalance reflects availability: short PRDs are abundant in public repositories, while full-length PRDs are rarely shared openly. Despite the imbalance, the corpus provides sufficient diversity to test whether the metrics produce distinguishable document profiles across the five categories.

B. Results

Welch's ANOVA test, which does not assume equal variances across groups, was performed for each of the 19 metrics across the five categories. All p-values were corrected for multiple comparisons using the Benjamini–Hochberg false discovery rate (FDR) procedure [22]. All 19 metrics were statistically significant after FDR correction ($p < 0.05$). As a non-parametric robustness check, the Kruskal–Wallis H-test confirmed significance for all 19 metrics.

Effect sizes were quantified using eta-squared (η^2). Entity coherence and structure metrics dominate the top positions: `sentences_analyzed` ($F = 156.7$, $\eta^2 = 0.49$), `lists_per_sentence` ($F = 101.3$, $\eta^2 = 0.32$), `headings_per_sentence` ($F = 96.6$,

$\eta^2 = 0.36$). The CPRE glossary metric illustrates a gap between statistical significance and practical effect: `cpre_terms_per_sentence` has the second-highest F-statistic ($F = 116.7$) but the lowest eta-squared ($\eta^2 = 0.01$), indicating statistical significance driven by large sample size rather than practical effect. This metric is retained because low discriminative power among document types does not mean it cannot predict extraction quality: a document's density of standard RE terminology may correlate with how well an LLM extracts requirements from it, regardless of which category the document belongs to. Stage 2 will test this directly.

A Random Forest classifier trained on all 19 metrics achieved 93.09% cross-validation accuracy (stratified 5-fold, `cross_val_predict`) in predicting document category. Per-class analysis reveals uneven performance: Short PRD (F1 $= 0.967$, $n = 225$) and Wikipedia (F1 $= 1.000$, $n = 13$) are classified near-perfectly, while Long PRD (F1 $= 0.588$, $n = 11$, recall $= 0.455$) is the weakest class—6 of 11 Long PRDs were misclassified, primarily as PURE (4 cases), reflecting shared structural properties. The gap between macro F1 (0.865) and weighted F1 (0.928) quantifies this small-class underperformance.

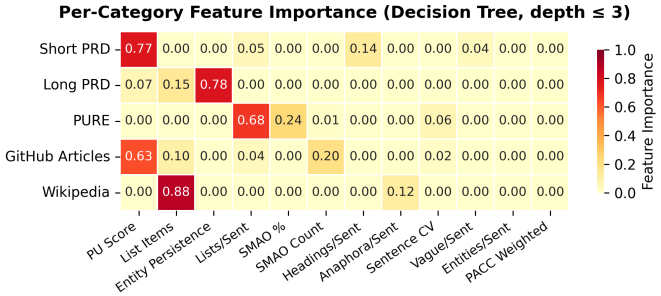

Fig. 1. Per-category feature importance from shallow decision trees (one-vs-rest, depth ≤ 3), trained without length-proxy metrics. Each row is a document category; each column is a metric. Darker cells indicate higher importance for that split. Each category relies on a distinct subset of metrics.

To identify which metrics characterize each category, we fit shallow decision trees (depth ≤ 3, one-vs-rest) excluding the two length-proxy metrics (`sentences_analyzed`, `total_entities`). Fig. 1 shows the resulting feature importance per category. Each category relies on a distinct metric subset: Short PRD is dominated by PU Score (0.77) and headings density; Long PRD by entity persistence (0.78); PURE by list density (0.68) and SMAO percentage; Wikipedia by list items (0.88) and anaphora. Notably, removing length proxies improves Short PRD classification ($F1 = 0.957$ vs. 0.931 with length), confirming that content-level metrics carry the discriminative signal independently of document size.

Dimension independence and metric redundancy (Spearman correlation). Correlation analysis reveals that `smao_percentage` exhibits low correlation with most other metrics, indicating that requirement-style writing constitutes an independent dimension—a document can be well-structured and readable without using formal

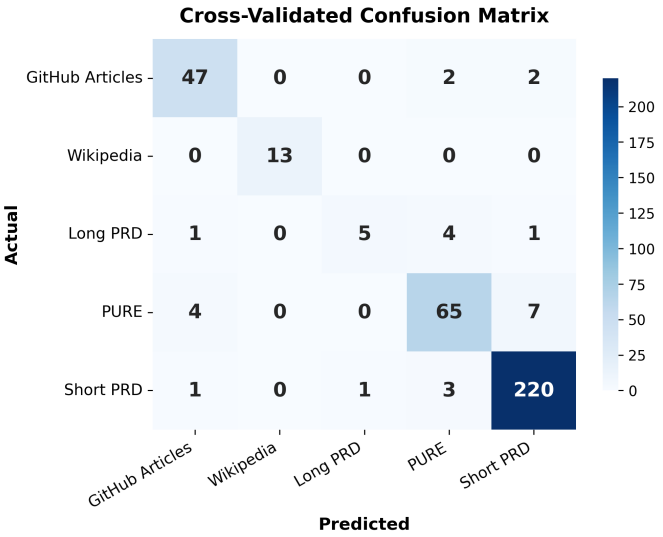

Fig. 2. Cross-validated confusion matrix. Rows = actual category, columns = predicted. Long PRD is primarily confused with PURE (4 of 6 errors). Wikipedia is perfectly classified.

requirement patterns. This independence is relevant for Stage 2: if requirements readiness predicts extraction quality independently of other dimensions, text difficulty is a multi-dimensional construct rather than a single scale.

However, substantial redundancy exists *within* dimensions. Among the entity coherence metrics, PW Score and PACC Uniform correlate at $\rho = 0.96$, and PU Score correlates with Sentences Analyzed at $\rho = 0.93$, meaning PU Score partially proxies for document length. SMAO Count and SMAO % correlate at $\rho = 0.98$, and the structure metrics Headings/Sent and Lists/Sent at $\rho = 0.79$. Cross-dimension correlations also emerge: Grade Level correlates with both CPRE/Sent ($\rho = 0.70$) and Headings/Sent ($\rho = 0.69$), suggesting that technically structured documents tend to use domain terminology and score higher on readability scales. A principal component analysis confirms that 10 components explain 90% of variance, confirming that the 19 metrics span approximately 10 independent axes rather than 19. Stage 2 will prune redundant metrics to avoid inflating the apparent dimensionality of the feature space.

C. Interpretation

These results confirm that the metrics produce distinguishable document profiles—a necessary precondition, not the end goal. The per-class breakdown is instructive. Long PRD's poor recall (0.455) shows that the current metrics cannot reliably separate long requirements documents from heterogeneous markdown content (PURE): per-sentence normalization removes length as a signal, so the two categories overlap in metric space. Conversely, Wikipedia's perfect classification despite only 13 documents confirms that its metric profile is genuinely distinctive—the metrics capture real text properties, not just sample-size artifacts. The macro/weighted $F1$ gap

quantifies the cost of class imbalance and will inform Stage 2 corpus design.

Document length alone is a trivially available predictor of extraction quality. The multi-dimensional metric framework adds value only if it predicts quality better than length alone. Stage 2 will use length as a baseline against which the incremental predictive power of the metric profiles is measured.

V. Research Roadmap

The preliminary evaluation establishes that the 19 retained metrics produce distinguishable document profiles. The central question of this research remains open: do these profiles predict how well an LLM can extract requirements from a given document? Stages 2 and 3 address this question through progressively broader experiments.

A. Stage 2: Correlating Metrics with Extraction Quality

Stage 2 constitutes the core experiment of this research program. The validated metrics will serve as independent variables; LLM extraction quality will serve as the dependent variable.

Corpus construction. The corpus used in Stage 1 validated the discriminative power of metrics, but it was not annotated with ground-truth requirements. Existing RE datasets typically contain isolated requirements (individual sentences or user stories) rather than full-length source documents paired with the requirements they contain [7]. Stage 2 requires a new corpus, or a significant extension of the existing one, with human-labeled requirements serving as the gold standard for evaluating extraction quality.

The target corpus comprises 100–200 documents spanning the full difficulty spectrum. Document sources will include: formal SRS and PRD documents from open-source projects and industry partners; meeting notes and brainstorming transcripts where requirements are embedded in conversational text; product backlogs and user story collections in varying levels of formality; regulatory and compliance documents where requirements are interspersed with explanatory text; and technical documentation where system capabilities are described but not explicitly stated as requirements.

A critical design decision is the document selection strategy. Documents will be sampled to ensure coverage across the metric space defined by the five dimensions. Specifically, the corpus will include documents that score high on some dimensions but low on others (e.g., high readability but low structure; high SMAO compliance but low entity coherence), avoiding a corpus where all metrics correlate uniformly. The Stage 1 metric profiles will guide this selection: documents will be chosen to fill gaps in the metric space that the Stage 1 convenience sample left uncovered.

Each document will be annotated for ground-truth requirements by at least two independent annotators following established RE annotation guidelines. The annotation schema will define what constitutes a requirement (functional, non-functional, and constraint), how to handle implicit requirements (stated indirectly or requiring interpretation), and how to delineate requirement boundaries when multiple requirements appear in a single sentence. Inter-annotator agreement will be measured using Cohen's kappa. Disagreements will be resolved through discussion, with a third annotator adjudicating where necessary. The annotated corpus will be made publicly available as a reusable benchmark for the RE community.

Extraction task. One or more LLMs will be prompted to extract requirements from each document using zero-shot prompting (no task-specific training or in-context examples). A standardized prompt template will be used to control for prompt engineering effects. The prompt will instruct the model to produce a structured list of requirements, each containing the requirement text, the source sentence or paragraph, and a confidence score. Extraction quality will be scored using precision, recall, and F1 against the human-labeled ground truth at the requirement statement level.

Analysis. For each document, the 19-metric profile will be computed and correlated with extraction quality scores. Both per-metric Spearman correlation and multivariate regression (including Random Forest feature importance) will be applied to identify which metrics (and which metric combinations) best predict extraction quality. Partial dependence plots will visualize how extraction quality changes as individual metrics vary while others are held constant.

Threshold detection. A key objective of Stage 2 is to identify threshold effects—specific metric values below (or above) which extraction quality degrades sharply. For example, there may exist a `consensus_grade_level` above which LLM extraction precision drops below an acceptable threshold, or an `entity_persistence_ratio` below which recall degrades. Identifying such thresholds would provide practitioners with actionable decision criteria for when automated extraction is likely to succeed.

Research questions for Stage 2:

- **RQ1:** Which text metrics are the strongest predictors of requirements extraction quality?
- **RQ2:** Do threshold effects exist—specific metric values below which extraction quality degrades sharply?
- **RQ3:** Can a document's metric profile predict extraction quality before running the LLM?

B. Stage 3: Multi-LLM Comparison and Failure Boundary Mapping

Stage 3 extends the single-LLM evaluation to a comparative study across multiple LLMs and prompting strategies, following the multi-model, multi-prompt evaluation design demonstrated for traceability by Alturayeif et al. [9]. The goal is to determine whether different models fail on different text dimensions, and whether the metric profiles can recommend the most suitable extraction technique for a given document.

Model selection. Stage 3 will evaluate a set of current-generation LLMs (e.g., GPT, Claude, Llama, Gemini) representing different model architectures (proprietary vs. open-weight, different parameter counts). The selection will be finalized at experiment time to use the most current models available, ensuring practical relevance of the findings.

Prompting strategies. Each model will be evaluated under multiple prompting strategies: zero-shot (baseline from Stage 2), few-shot (providing 2–5 annotated extraction examples), and chain-of-thought (instructing the model to reason about document structure before extracting). Reasoning-augmented models (e.g., o1-class architectures) and task-specific fine-tuning may also be considered if Stage 2 results suggest that prompting alone cannot compensate for high text difficulty. This design separates model capability from prompt engineering effects and tests whether more sophisticated prompting can compensate for text difficulty.

Failure boundary mapping. The central analytical goal is to map technique-specific failure boundaries in the metric space. For each model–prompt combination, regression models will identify the metric regions where extraction quality degrades below a predefined threshold (e.g., F1 < 0.5). Comparing these boundaries across models will reveal: (a) whether some models are more robust to specific text properties (e.g., low coherence, high readability grade level), (b) whether failure boundaries shift with prompting strategy, and (c) whether complementary model selection is possible—assigning documents to the model best suited for their metric profile.

Practical outcome. The envisioned deliverable of Stage 3 is a decision support tool: given a document's metric profile, the tool recommends which model–prompt combination is most likely to produce acceptable extraction quality, or flags the document as requiring manual elicitation. This tool would operationalize the research findings for RE practitioners.

Research questions for Stage 3:

- **RQ4:** Do different LLMs degrade on different text dimensions?
- **RQ5:** Can technique-specific failure boundaries be mapped—for each model–prompt combination, which metric profiles cause extraction quality to degrade?
- **RQ6:** Given a document's metric profile, can the most suitable extraction technique be recommended?

C. Risks and Mitigation

TABLE II
RESEARCH RISKS AND MITIGATIONS

Risk	Mitigation
Metrics do not correlate with extraction quality	Negative result is informative; explore alternative characterizations
Ground-truth annotation disagreement	Multiple annotators; inter-annotator agreement; established guidelines
Corpus too homogeneous	Expand with industry, standards, and informal sources
LLM version drift	Record exact model versions and API parameters
Weak individual but strong combined correlations	Multivariate analysis alongside per-metric correlation
Prompting confounds model comparison	Standardize prompts; test multiple strategies per model

VI. Discussion and Implications

A framework for controlled evaluation. Current LLM evaluations in RE typically report aggregate performance on benchmark datasets without characterizing the input. If the hypothesis holds, the metrics toolkit provides a method for controlling for input difficulty when evaluating any NLP-based RE technique. Researchers could report not just "Model X achieved F1 $= 0.85$" but "Model X achieved F1 $= 0.85$ on documents with median `smao_percentage` of 12% and `entity_persistence_ratio` of 0.45." Characterizing the input would make results more comparable across studies.

Multi-dimensional difficulty. The independence of `smao_percentage` from structure and readability metrics in the preliminary evaluation suggests that text difficulty for requirements extraction is not a single scale but a multi-dimensional space. A document may score well on readability and structure yet contain no formal requirement patterns—or vice versa. This multi-dimensionality has practical implications: a single "difficulty score" would obscure the specific text properties that cause extraction to fail. Stage 2 will test whether different dimensions predict different aspects of extraction quality (e.g., structure predicts recall while requirements readiness predicts precision).

Toward model-aware document triage. If Stage 3 confirms that different LLMs fail on different dimensions, the practical implication is that no single model is universally optimal. Instead, practitioners could use the metrics toolkit as a triage step: route formal SRS documents to a model optimized for structured input, and brainstorming transcripts to a model more robust to low coherence and informal language. The failure boundary maps from Stage 3 would provide the empirical basis for such routing decisions.

Interpretable features vs. embedding-based prediction. A deliberate design choice of this work is the use of interpretable metrics rather than embedding-based features (e.g., document embeddings from a pre-trained language model fed into a classifier). Embedding-based approaches would likely yield higher classification accuracy on the Stage 1 task, but would not serve the research objective. The goal is not to predict document category—it is to identify which specific text properties correlate with extraction difficulty, so that practitioners can act on the findings (e.g., "restructure this document by adding headings" or "replace vague terms before running extraction"). Stage 2 may also include an embedding-based baseline to quantify the accuracy gap and assess whether the interpretable metrics capture sufficient variance for practical prediction.

Surface metrics and LLM processing. LLMs do not process text through the same cognitive mechanisms that readability formulas model (e.g., working memory load from long sentences). This creates a construct validity question: do metrics designed for human reading difficulty predict machine extraction difficulty? We treat this as an empirical question rather than an assumption. Stage 1 establishes that the metrics discriminate among document types; Stage 2 will test whether

they also predict LLM extraction quality. If the correlation is weak, this would indicate that LLM-specific difficulty dimensions (e.g., tokenization artifacts, positional encoding limitations, attention span effects) must supplement or replace the human-oriented metrics. The three-stage design explicitly accommodates this possibility: Stage 2 results will determine whether the current metric set is sufficient or requires revision before Stage 3.

Broader applicability. While this research focuses on requirements extraction, the approach of characterizing input difficulty as an independent variable may generalize to other RE tasks: ambiguity detection, requirements classification, completeness checking, and to NLP tasks beyond RE. Any task where model performance varies with input properties could benefit from a systematic difficulty characterization.

Open questions for the RE community. We invite discussion on several avenues: (1) Are the five dimensions sufficient to characterize extraction difficulty, or do important text properties remain uncaptured (e.g., domain-specific terminology density, EARS pattern compliance, cross-reference complexity)? (2) The SMAO detector currently targets IEEE 29148-style "shall" statements. Agile formats, such as user stories ("As a… I want…"), Gherkin/BDD scenarios, acceptance criteria, and others have fundamentally different syntactic structures and would require dedicated pattern detectors rather than extensions of the current one. We consider this a separate study direction. (3) How should the toolkit be adapted for multilingual requirements artifacts or documents that intermix natural language with code, models, or diagrams?

VII. Limitations

English only. All word lists, readability formulas, and pattern matching assume English text. The toolkit does not handle multilingual requirements artifacts.

Heuristic-based analysis. Four of five analyzers rely on pattern matching rather than deep NLP. SMAO detection uses spaCy dependency parsing with regex fallback, which may miss non-standard requirement formulations.

No coreference resolution. Anaphoric pronouns are counted but not resolved to their referents. A document containing many pronouns that all refer to the same entity is scored identically to a document with genuinely ambiguous references.

Only form, but not meaning. The toolkit measures surface-level text properties. A well-structured requirements specification containing contradictory or incorrect requirements would score well on structural and readability metrics.

Corpus limitations. The preliminary evaluation uses a convenience sample of 376 documents with imbalanced category sizes.

Proof-of-concept scope. Stage 1 validates that the metrics discriminate between document types. The link between metric-based document characterization and actual LLM extraction quality remains an untested hypothesis at this stage.

VIII. Conclusion

This paper presented a research program to map the boundaries of LLM-based requirements extraction by characterizing input text difficulty as an independent variable. The program proceeds in three stages: metric definition and validation, correlation with extraction quality, and multi-LLM comparison.

As an initial step, we developed a toolkit computing 19 metrics across five dimensions (entity coherence, readability, structure, requirements readiness, and domain terminology coverage) for statistical analysis. A preliminary feasibility validation on a 376-document corpus confirms that these metrics capture meaningful variation: all 19 metrics significantly discriminate between document categories (Welch's ANOVA, Benjamini–Hochberg FDR correction, $p < 0.05$), and the metric profiles achieve 93.09% classification accuracy with macro F1 $= 0.865$.

The core contribution of this research preview is not the toolkit itself but the research approach: treating text difficulty as measurable and controllable when evaluating LLM-based RE techniques. Stage 2 will test whether these metrics predict extraction quality for individual LLMs. Stage 3 will extend the evaluation to multiple models and prompting strategies to map technique-specific failure boundaries. The envisioned outcome is a practical mapping from document characteristics to expected extraction quality, enabling practitioners to make informed decisions about when automated extraction is appropriate and which technique to apply, and providing researchers with a controlled framework for comparing RE tools on a common difficulty scale.

Data Availability Statement

The metrics toolkit source code, the 376-document corpus, and all analysis scripts are publicly available at https://doi.org/10.5281/zenodo.20709074.

Acknowledgments

AI-Generated Content Disclosure: An AI coding assistant was used during development of the metrics toolkit and in drafting this manuscript. The AI system assisted with Python code generation for the five analyzers and the statistical analysis pipeline, and with iterative editing of the manuscript text for clarity and structure. All research design decisions, metric selection, corpus construction, result interpretation, and final manuscript writing were performed by the authors.

References

[1] S. Abualhaija, C. Arora, M. Sabetzadeh, L. C. Briand, and M. Traynor, "Automated demarcation of requirements in textual specifications: a machine learning-based approach," *Empirical Software Engineering*, vol. 25, no. 6, pp. 5454–5497, Nov. 2020. [Online]. Available: https://doi.org/10.1007/s10664-020-09864-1

[2] C. dos Santos, K. Bouchard, and B. Napoleão, "Automatic user story generation: a comprehensive systematic literature review," *International Journal of Data Science and Analytics*, vol. 20, pp. 1–24, Jun. 2024.

[3] A. Korn, S. Gorsch, and A. Vogelsang, "LLMREI: Automating requirements elicitation interviews with LLMs," in *2025 IEEE 33rd International Requirements Engineering Conference (RE)*, 2025, pp. 19–30.

[4] S. Bashir, A. Ferrari, A. Khan, P. E. Strandberg, Z. Haider, M. Saadatmand, and M. Bohlin, "Requirements ambiguity detection and explanation with llms: An industrial study," in *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2025, pp. 620–631.

[5] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, and R. T. Batista-Navarro, "Natural language processing for requirements engineering: A systematic mapping study," *ACM Comput. Surv.*, vol. 54, no. 3, Apr. 2021. [Online]. Available: https://doi.org/10.1145/3444689

[6] A. Ferrari, L. Zhao, and W. Alhoshan, "Nlp for requirements engineering: Tasks, techniques, tools, and technologies," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2021, pp. 322–323.

[7] M. A. Zadenoori, J. Dąbrowski, W. Alhoshan, L. Zhao, and A. Ferrari, "Large language models (llms) for requirements engineering (re): A systematic literature review," 2025. [Online]. Available: https://arxiv.org/abs/2509.11446

[8] C. Arora, J. Grundy, and M. Abdelrazek, "Advancing requirements engineering through generative AI: Assessing the role of LLMs," in *Generative AI for Effective Software Development*. Springer, 2024, pp. 129–148.

[9] N. Alturayeif, I. Ahmad, and J. Hassine, "Tracellm: leveraging large language models with prompt engineering for enhanced requirements traceability," *Requirements Engineering*, vol. 31, no. 1, May 2026. [Online]. Available: http://dx.doi.org/10.1007/s00766-026-00460-1

[10] I. Darif, F. Niu, M. Abdellatif, L. C. Briand, R. S., and A. Adiththan, "Automating the detection of requirement dependencies using large language models," 2026. [Online]. Available: https://arxiv.org/abs/2602.22456

[11] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, no. 3, pp. 221–233, 1948.

[12] R. Gunning, *The Technique of Clear Writing*. New York: McGraw-Hill, 1968.

[13] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel," Chief of Naval Technical Training, Millington, TN, Tech. Rep. Research Branch Report 8-75, 1975.

[14] G. A. Harry and M. Laughlin, "Smog grading - a new readability formula." *The Journal of Reading*, 1969. [Online]. Available: https://api.semanticscholar.org/CorpusID:9571753

[15] R. Barzilay and M. Lapata, "Modeling local coherence: An entity-based approach," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, K. Knight, H. T. Ng, and K. Oflazer, Eds. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 141–148. [Online]. Available: https://aclanthology.org/P05-1018/

[16] D. A. Palma, C. M. Soto, M. Veliz, B. Karelovic, and B. Riffo, "TRUNAJOD: A text complexity library to enhance natural language processing," *Journal of Open Source Software*, vol. 6, no. 60, p. 3153, 2021.

[17] ISO/IEC/IEEE, "Systems and software engineering – life cycle processes – requirements engineering," International Organization for Standardization / Institute of Electrical and Electronics Engineers, Geneva, Switzerland / New York, NY, USA, Standard, November 2018. [Online]. Available: https://standards.ieee.org/ieee/29148/12262

[18] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in *IEEE International Requirements Engineering Conference*, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:34495464

[19] M. Glinz, "Glossary of requirements engineering terminology," International Requirements Engineering Board (IREB e.V.), Tech. Rep., 2024, version 2.2. [Online]. Available: https://cpre.ireb.org/en/downloads-and-resources/glossary

[20] P. S. Kummler and H. Fromm, "A closer look on the difficulties to determine the quality of software requirements," in *Software Engineering 2019, Fachtagung des GI-Fachbereichs Softwaretechnik*, ser. Lecture Notes in Informatics (LNI). Gesellschaft für Informatik, 2019.

[21] A. Ferrari, G. O. Spagnolo, and S. Gnesi, "PURE: A dataset of public requirements documents," in *Proceedings of the 25th IEEE International Requirements Engineering Conference (RE '17)*. IEEE, 2017, pp. 502–505.

[22] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995. [Online]. Available: http://www.jstor.org/stable/2346101