大規模言語モデルによる要求仕様書の品質評価

鈴木淳[2] 村瀬文彦[1] 水野伸洋[3] 髙木理恵子[2] 不破慎之介[2] 塚原裕史[3] 中江俊博[1]
[1]株式会社デンソー [2]株式会社デンソークリエイト [3]株式会社デンソーアイティーラボラトリ
atsushi.suzuki.j3h@jpgr.denso.com

概要

車載ソフト開発を高品質かつ低コストに実現するために，人手による作業をできる限り自動化することが急務となっている．本研究では特に要求仕様書の品質評価 (要求インスペクション) において大規模言語モデル (LLM) による自動化が可能であるかを具体的な事例によって分析した．分析対象として過去に人手で行われたインスペクション結果を用いて，どの程度それらの結果を再現することができるかという再現率及び人手では見逃されていた問題箇所がどれくらい漏れなく指摘できるかという適合率を評価し, LLM 活用の有効性と課題を分析した．その結果, LLM 単独では必ずしも十分な精度を得ることができないが, 正規表現や形態素解析などのルールベース処理や従来の自然言語処理による前後処理と組合わせることで精度を向上できる示唆を得た．

1 はじめに

自動運転, コネックティッド・カー，電動化など SDV(Software Defined Vehicle) によるクルマの進化により自動車の開発におけるソフト開発の比重は年々増加しており，2030 年には開発工数の半分がソフト開発になるとも予測されている[1)]．高品質なソフト開発を効率的かつ低コストに実現するために，要求仕様を明確に定義し, 管理する要求エンジニアリング (Requrements Engineering) [1] が重要となっている．デジタル・トランスフォーメイション (DX) の推進によって, 要求仕様書のデジタル化はほぼ達成されているものの, その形式はメーカーや製品ごとに独自のフォーマットとなっている．また，現場における慣習などの暗黙ルールなどによる担当者間で解釈の不一致や自然言語による記述による曖昧さや表記揺れにより仕様が不明確となり, 後工程において問題が発覚することが頻発している [2]．このように要求仕様段階で見過ごされた問題の修正には, その原因や影響範囲の把握及び修正対応など手戻りのコストが非常に高く，要求仕様段階において早期に修正することが非常に重要となる．そのためには要求仕様書についても設計文書における UML や SysML などのように標準化された記述モデルとその作成支援ツールの活用が望まれるが，車載ソフトにはそれぞれに特有の要求や制約があり, 現状では十分に実運用できていない．

これらの要因から要求インスペクションの自動化が困難なため，人手によるインスペクションにより要求仕様書それぞれの違いを埋め合わせなければならず, コストが高いと共に問題箇所の見逃しというミスも発生するという問題がある．さらに評価基準が担当者の知識や経験によって変わってしまう恐れもある．これらの問題を解決するために要求インスペクションにおける評価基準の標準化 [3] や形態素解析などの従来の自然言語処理手法により要求インスペクションを部分的に自動化する研究が進められてきた [4, 5]．現在, LLM によってさまざまな自然言語処理におけるタスクを汎用的に遂行可能になってきたことで，これまで要求エンジニアリングにおいて部分的に検討されてきた自動化を一気通貫に実行できる可能性が高まってきている [6, 7, 8, 9]．特に要求仕様書の形式の違いに対して, 従来の自然言語処理では個別に対応する必要があったが，LLM であれば形式の違いを吸収し，汎用的に対応できるのではないかと期待できる．

本研究では要求仕様書の表現品質の記述ガイドライン [3] に従った要求インスペクションを LLM に適用し, どの程度自動化を実現することが可能であるかを検討することを目的とする．そのために, 過去に実際の製品において人手で行った要求インスペクション結果と LLM で行った結果との比較を行う．

1) https://www.nam.co.jp/market/column/analyst/2022/220921.html

2　関連研究

要求インスペクションの自動化に関連する先行研究として，従来からの自然言語処理を応用したものとして要求仕様書における曖昧な表現や表記揺れの検出に関しては形態素解析を用いる手法などが提案されている [4, 5].

近年は LLM を適用する手法も多く提案されており，例えば, BERT を用いてトラブル報告書から現象や処置の情報を認識させて, 様々なタスクへ応用する研究 [10]，システム要求仕様書における非機能要求を分類する研究 [11] やトラブル報告書におけるフィールド分類を行う研究 [10] がある．また，GPT-4 及び CodeLlama による要求仕様生成と与えられた要求仕様書における問題箇所の指摘の自動化について人手による作業結果との比較を行った研究がある [9]．その結果，要求仕様書生成については新人エンジニアと同等レベルの品質のものが生成でき, 問題箇所の指摘については GPT-4 であれば正しい指摘とフィードバックを与えることが可能であると評価がされている．

3　要求インスペクションの概要

本研究における要求インスペクション・タスクは，以下に述べるソフトウェア要求仕様書 (Software Requirements Specification, 以下 SRS と記す) 記述ガイドラインに基づき, 要求仕様書において SRS 記述ガイドラインの適合基準を満たさない箇所を検出し，その理由及び修正案を提示することである．

3.1　SRS 記述ガイドライン

SRS 品質特性とは要求仕様書が満たすべき条件や必要な章・節構成を定義したものであり，ISO/IEC/IEEE によって基本となるものがまとめられている [12]. 章・節構成などの構造的な条件に加えて，良い要求仕様書が満たすべき文章表現や記述内容に関する条件も含まれている．これらの品質特性は大きく表現品質と内容品質の 2 つのグループに分類することができる [13]．SRS 記述ガイドラインは，各章・節構成や満たされるべき品質特性の必要性及びそれらの項目に書きべき内容や誤りを防止するために注意すべき点などの事例を加えることで, 要求仕様書の記述を助ける役割を持つものである [3].

表 2 は我々が使用する SRS 記述ガイドラインの品質特性項目の一部である．品質特性は大きく 11 項目のカテゴリに分類されており，本研究ではそれらを表現品質と内容品質に分け，かつ各品質特性の適合判定を行うための検査ルールを設定した．各検査ルールにはその適合基準を与えてある．さらにこれらの各検査ルールに対しては具体的な誤り事例を付与してあるが，具体的に示すことは省略する．

表現品質　表現品質は文書構成や文章のシンタックスに関する品質を定める項目のカテゴリである．たとえば, 表記揺れ, 記入漏れ (表や図番号，凡例の説明など), 参照 ID の間違いなどが典型的な表層的な誤りである．このような表層的な品質に加えて，形式的に判定が可能なレベルにおける場合分けの完全性などの項目もある．さらに，日本語特有の助詞の使い方により意味が曖昧となることを防止するための項目も含まれる．

内容品質　内容品質は文章の表層的あるいは形式的な分析では判定ができないセマンティクスに関わる品質を定める項目のカテゴリである．機能の定義に実装に係る内容などが記述されていないか，インタフェース要求の不整合がないか，要求を検証する妥当な方法があるかなどの項目がそれに当たる．

4　処理プロセス

図 1 に，要求インスペクションの処理プロセスの概要を示す．処理プロセスは大きく要求仕様書から検査対象の文を抽出する文抽出処理，抽出した文について品質検査を行うインスペクション処理及び検査結果を出力するレポート生成処理から成る．以上の処理プロセスをクラウド上に実装し，ブラウザをユーザインタフェースとして要求インスペクションを行えるようにした．

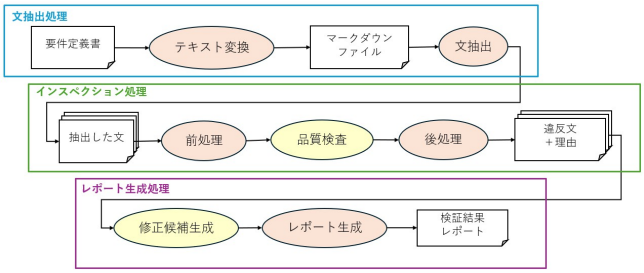

図 1　要求インスペクション処理プロセス (LLM による処理が主となる部分を黄色で示す)

4.1　文抽出処理

要求仕様書は主に Word や PowerPoint などで作成されており，章や節構成はあるものの内容は自然言

語で記述された構造化されていない文章のみからなる場合や，表形式で記述されていてある程度構造化はされているが，その表の各セルの中身はまた文章による記述やさらに表や図が入れ子になっているものもあるなど，多種多様な形式になっていることを想定する必要がある．

LLM による要求インスペクションでは，このような形式の違いを LLM 自身が把握し，形式ごとに適切に情報を把握し，品質検査を行うことができることを期待している．

テキスト変換　要求仕様書のファイル形式の違いを吸収するために，まずテキストへ変換する．本研究ではテキスト変換については Azure AI Document Intelligence[2] (ADI) を利用する．

要求仕様書には図表データが多く含まれているため，それらの情報をテキストとして構造を保ったまま抽出する必要がある．従って単純にテキスト化するのではなく，出力形式として HTML, JSON あるいはマークダウンなどの構造を記述できる形式にテキスト化する必要がある．HTML であれば入れ子になった表でもその構造を保ったまま出力可能であるが，タグによってテキストが冗長となる懸念がある．一方，JSON では構造の解釈に曖昧さが生じる．そこで今回は比較的タグなどのオーバーヘッドが少ないが，表などの構造を表現可能なマークダウン形式へ変換する．また，マークダウン形式を指定することで性能が改善する場合があるという指摘もあり [14], 好ましいと判断した．

但し, 今回の評価に用いた要求仕様書によっては表が ADI によって正しくパースされず，表の構造が崩れて出力されることがあった．また，表が入れ子になっている場合にも入れ子の表は表として認識されないなどの問題があった．図についても表以上に読み取り内容が不十分なため，これらの部分については，今回の評価対象から除外することにした．

文抽出　要求仕様書は一般に数十から数百ページに渡って記述されており，マークダウン形式に変換されたテキストの全体長は非常に大きくなる．しかし，品質特性項目によっては要求仕様書の局所的な文のみで検査可能なものも多い．従って，まずテキスト全体を節へ分割し，各節の中のテキストを文の単位に分割したうえで，各品質特性項目へ要求仕様書の内容を渡すようにする．

4.2　インスペクション処理

インスペクション処理においては，要求仕様書から抽出された文をそのまま LLM へ渡して品質検査を行うだけではなく，その前後に他の処理を入れて，LLM による処理を補うことも考える．

前処理　評価対象の文を抽出後，LLM による品質検査を行う前に必要な処理を行う．マークダウン形式のテキストには章・節構成や表構造を表現するためのマークが含まれている．たとえば要求 ID，定義文，制約などの内容が表形式で記述されている場合，それらの記述場所を示すためのヘッダー部などのテキストも含まれている．そこでこのような要求仕様書の記述フォーマットをメタ情報として LLM へ教示するか，ヘッダー情報部と要求を定義している内容部の文とを分類して抽出し，内容部の文について品質検査を行えるようにする必要がある．例えば，機能定義の曖昧性を評価する場合には，機能定義を記述している部分を要求仕様書全体の文の中から分類して抽出する．あるいは正規表現処理でさらにテキストを加工する，形態素解析によって補助的なラベル付けを行うなどの処理を行う．

品質検査　前処理によって加工された文に対して品質検査を LLM によって行う．本論文では LLM として Azure OpenAI Service [3] の GPT-4o モデルを使用する．

無曖昧性の品質検査では適合基準を指示内容に与え，違反事例をいくつか教示する few-shot プロンプトによる処理を行う．

完全性，変更容易性，単独性の品質検査では節単位で品質検査を行う．適合基準を指示内容に与え，さらに前処理で判定された表の構造に関するメタデータ情報を few-shot table understanding [15] に準ずる手法で与えたプロンプトを用いた．

後処理　LLM による品質検査結果をパースして，品質違反があった文やその違反内容を抽出する．またはその出力内容をさらに処理して最終的なインスペクション結果を判定する．

4.3　レポート生成処理

修正候補生成　品質違反が検出された部分と違反内容を参照し，LLM によって修正候補を生成する．但し，無曖昧性の品質検査においては違反箇所の検

2）https://azure.microsoft.com/ja-jp/products/ai-services/ai-document-intelligence

3）https://azure.microsoft.com/ja-jp/products/ai-services/openai-service

出と同時に修正候補も生成させている．

出力生成　違反箇所，違反内容及び修正候補の内容をユーザインタフェース画面における表示形式に合わせて整理する．

5　評価実験

今回は表 2 の品質特性の中から，無曖昧性，完全性，変更容易性，単独性に関わる 6 つの検査ルールを LLM へ教示して, 要求インスペクションの自動化への有効性を分析する．

5.1　評価データ

車載の駆動系ソフトウェア製品に関する要求仕様書を評価データとして使用した．本仕様書は全体で 73 ページあり，その内の 52 ページに要求が記載されており，人手による要求インスペクションで指摘された品質違反の箇所及び内容，どのように修正すべきかという結果が付属されている．この指摘結果を正解データと見なし，自動インスペクション結果の再現率を評価する．また自動インスペクション結果の妥当性を人手で判定し，適合率を評価する．LLM による指摘結果において，指摘箇所と指摘内容及び修正候補が人手指摘と同じ内容であった場合に正しく指摘ができていると判定する．なお人手によるインスペクション結果がない別製品の要求仕様書 (全 111 頁中，要求が 85 頁記載) についても人手により適合率を評価する．

5.2　評価結果

評価結果を表 1 に示す．検査ルールとして 1) 簡潔な文, 2) 係り受け, 3) 誤字脱字, 4) 完全な場合分け, 5) 誤った記載場所, 6) 単一の要求による品質検査を行った (各検査ルールの内容については表 2 を参照)．インスペクション処理において，要求仕様書におけるテキストをそのまま LLM へ検査対象のテキストとして入力した場合，適合率と再現率ともに低い値となった．例えば，専門用語を誤字と判断するケースなどが多く見られた．つまり，車載ソフトウェア製品の要求仕様書のようにドメイン独自の用語や表現が多くある場合には，汎用的な LLM 単独では few-shot プロンプトにより検査事例を与えたとしても，高精度の品質検査を実現するには不十分であると考えらえる．

LLM を単独で使用するのではなく，各要求仕様書のデータについて，前処理において LLM が品質検査を的確に行うことができるようにするために，注意する部分や手がかりとなる情報を付加することが必要であると考えた．そこで，1) 簡潔な文, 2) 係り受けの検査ルールについては, 前処理として GiNZA [16] により形態素解析を行い，それらと 3) 誤字脱字については，出力に対してルールベースの後処理により品質違反候補を検出し，その部分について品質検査を LLM に行わせるようにすることで大きく性能が向上することが分かった．誤字脱字については再現率が低いものの適合率はそれに比べて高く，人手で見逃している事例を漏れなく見つけられていると考えられる．また，4) 完全な場合分け, 5) 誤った記載場所, 6) 単一の要求の検査ルールについては，前処理として要求仕様書のフォーマットが予め定義されているフォーマットのどれに近いかを LLM によって推定し, そのフォーマット情報を付与して品質検査を LLM に行わせることが有効であることがわかった．なお人手評価がない別製品の要求仕様書についても本プロセスを適用し，高い適合率が得られている．

表 1　評価結果

SRS品質特性名	検査ルール名	人手指摘の結果あり (LLMのみ)		人手指摘の結果あり (LLM+ルールベース)		人手指摘の結果なし (LLM+ルールベース)
		適合率	再現率	適合率	再現率	適合率
無曖昧性	簡潔な文	**0.31**	**0.71**	**0.97**	**1.00**	**1.00**
	係り受け	**0.53**	**0.31**	**0.94**	**0.98**	**1.00**
完全性	誤字脱字	**0.47**	**0.30**	**0.78**	**0.47**	**0.79**
	完全な場合分け	**0.45**	**0.57**	**0.77**	**0.85**	**0.89**
変更容易性	誤った記載場所	**0.64**	**0.63**	**0.80**	**0.63**	**1.00**
単独性	単一の要求	**0.33**	**0.00**	**0.67**	**1.00**	**0.70**

6　おわりに

人手による要求インスペクション結果と比較し，現在の汎用的な LLM 単独による要求仕様書の品質検査では十分な精度は得られないが，前処理において要求仕様書から重要な箇所を抽出あるいは記述フォーマットの推定結果を付与するなど LLM へ与える入力の拡張することで精度が大きく向上できることを示唆する結果が得られた．今後は要求仕様書を単にマークダウン形式に変換するのではなく，要求仕様書のメタモデル [17] に基づき構造化したものを LLM へ入力することで，精度向上を進める．また LLM 拡張エージェント [18] により，高度な判定や自己検証を行えるような拡張を検討する．

参考文献

[1] Riad Sonbol, Ghaida Rebdawi, and Nada Ghneim. The use of NLP-based text representation techniques to support requirement engineering tasks: A systematic mapping review. **IEEE Access**, Vol. PP, pp. 1–1, 2022.

[2] 情報処理推進機構 (IPA). 2012 年度「ソフトウェア産業の実態把握に関する調査」 調査報告書, 2012. https://www.ipa.go.jp/digital/software-survey/kumikomi/hjuojm000000lps0-att/000026799.pdf.

[3] 不破慎之介, 蛸島昭之, 山田ひかり. 要求仕様書の記述を支援する記述ガイドの作成と評価. **JISA quarterly : bulletin**, No. 132, pp. 76–79, 2019.

[4] 柏原一雄, 不破慎之介, 石川冬樹, 長井亘, 林香織, 栗田太郎. 要求仕様の誤解釈を検出する Domain Word Modeling の提案. ソフトウェア・シンポジウム 2020, 6 2020.

[5] 柏原一雄, 新留光治, 李路雲, 鈴木淳, 松原潤弥, 不破慎之介. 要求仕様に対する形態素ベースドレビューの提案. ソフトウェア・シンポジウム 2021, 6 2021.

[6] Junda He, Christoph Treude, and David Lo. LLM-based multi-agent systems for software engineering: Vision and the road ahead. **ArXiv**, Vol. abs/2404.04834, , 2024.

[7] Maria Bonner, Marc Zeller, Gabor Schulz, and Ana Savu. LLM-based approach to automatically establish traceability between requirements and mbse. **INCOSE International Symposium**, 2024.

[8] Johannes J. Norheim, Eric Rebentisch, Dekai Xiao, Lorenz Draeger, Alain Kerbrat, and Olivier de Weck. Challenges in applying large language models to requirements engineering tasks. **Design Science**, 2024.

[9] Madhava Krishna, Bhagesh Gaur, Arsh Verma, and Pankaj Jalote. Using llms in software requirements specifications: An empirical evaluation. **2024 IEEE 32nd International Requirements Engineering Conference (RE)**, pp. 475–483, 2024.

[10] 労瑛瑩, 山崎智弘, 伊藤雅弘. トラブル報告書に特有のフィールド情報を用いた BERT の追加学習. 言語処理学会 第 29 回年次大会, pp. 311–316, 2023.

[11] 向田和弘, 福居誠二, 長岡武志, 北川貴之, 小形真平, 岡野浩三. 大規模言語モデルの Function Calling の構造化機能を活用した情報システム非機能要求の自動分類. 第 38 巻, 2024.

[12] C/S2ESC Software Systems Engineering Standards Committee. Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering, 2018. https://standards.ieee.org/ieee/29148/6937/.

[13] 不破慎之介, 山田ひかり, 蛸島昭之. 要求記述のスキル不足に対する SRS 記述ガイドの有効性評価. ソフトウェア・シンポジウム 2018, 6 2018.

[14] 伊東和香, 佐藤美唯, 志歩, 倉光君郎. コード生成タスクにおけるプロンプトの指示形式の差異が与える性能分析. 人工知能学会全国大会論文集, Vol. JSAI2024, pp. 4Xin243–4Xin243, 2024.

[15] Ruixue Liu, Shaozu Yuan, Aijun Dai, Lei Shen, Tiangang Zhu, Meng Chen, and Xiaodong He. Few-shot table understanding: A benchmark dataset and pre-training baseline. In **Proceedings of the 29th International Conference on Computational Linguistics**, pp. 3741–3752. International Committee on Computational Linguistics, October 2022.

[16] 松田寛. GiNZA - universal dependencies による実用的日本語解析. 自然言語処理, Vol. 27, No. 3, pp. 695–701, 2020.

[17] 不破慎之介, 小林展英. Next Design を用いた DX 推進ツールの構築プロセスの提案. 人工知能学会第二種研究会資料, Vol. 2023, No. KSN-033, p. 04, 2023.

[18] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. Augmented language models: a survey. **Trans. Mach. Learn. Res.**, Vol. 2023, , 2023.

A　SRS 記述ガイドライン

以下に, SRS 記述ガイドラインの品質特性項目の一部を抜粋したものを示す.

表 2　SRS 記述ガイドラインの品質特性項目

SRS品質特性名	分類	品質特性名	検査ルール名	適合基準
無曖昧性	表現品質	無曖昧性	簡潔な文	要求文に以下の表現を使わず記述していること 受動態，使役表現，二重否定，部分否定
			係り受け	以下のような表現を使わずに記述すること ・助詞「は」 ・同じ助詞の文節が連続 ・長文だが読点「、」なし
変更容易性	表現品質	変更容易性	誤った記載場所	記載すべき領域に記載していること
単独性	表現品質	単独性	単一の要求	一つの要件で複数の要求を記述していないこと
完全性	表現品質	記述の完全性	完全な場合分け	条件の抜け漏れ,ダブりが要件内にないこと
			誤字脱字	誤字脱字がないこと （スペルミス，漢字変換ミス，助詞誤り/欠如）
	内容品質	インターフェース要求の完全性	インターフェース要求の不足	入力文書で定義されているインターフェースが漏れなく定義されていること
限定性	表現品質	目的の明確性	要求の根拠	要求の根拠（目的または理由または動機または背景)が明確になっていること
	内容品質	限定性	不要なインターフェース要求	不要なインターフェース要求がないこと
実装独立性	表現品質	実装方法の独立性	実現方法の独立性	要求の実現手段を記述していないこと
	内容品質	実装制約の妥当性	実装制約の妥当性	機能要求とインターフェース要求に関連する制約が、設計・実装に対する制限を課していないこと
追跡性	表現品質	追跡可能性	要求発生源への追跡性	当該要求が上位要求と関連付けされていること
	内容品質	工程間の一貫性	トレーサビリティの妥当性	上位要求とソフトウェア要求間のトレーサビリティがが整合すること
一貫性	表現品質	記述の一貫性	表記揺れ	表記揺れがないこと（運転者/ドライバーなど）
	内容品質	機能要求と技術要求の一貫性	機能要求と制約の不整合	機能要求が制約を満たすこと
実現可能性	内容品質	実現可能性	実現可能性	開発制約の中で達成可能な要求であること
検証可能性	表現品質	検証方法の明確性	検証方法の明確性	各要求に検証方法を明記していること
	内容品質	検証可能性	検証可能性	要求が検証可能、かつ、要求の検証方法が妥当な方法であること
優先順位付け	表現品質	優先順位付け	機能の開発優先度	ソフトウェアの機能(または要求)毎に、開発の優先度(またはマイルストーン)とその根拠を記述していること