International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 6
ISSN: 2088-5334

Ambiguity Detection and Improvement for Malay Requirements Specification: A Systematic Review

Mohd Firdaus Zahrin [a], Mohd Hafeez Osman [a,*], Sa'adah Hassan [a], Azlena Haron [b], Alfian Abdul Halin [a]

[a] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Malaysia*
[b] *Information Management Division, Ministry of Education, 62604 Putrajaya, Malaysia*
*Corresponding author: *hafeez@upm.edu.my*

***Abstract*— Malaysian public sectors have invested billions in digitizing systems. Electronic government efforts created much software. Our informal interview taught us that many software projects encountered delays, and several failed. One of the main contributions of software failure is ambiguity in requirements specification (RS). Ambiguity is a familiar requirement smell that causes misinterpretation. Thus, we seek to devise a technique for detecting and improving ambiguous RS in the Malaysian public sector. One of our challenges is that the Malaysian public sector RS is developed in Malay, and most available techniques support English and other major languages. Hence, this paper investigates the automated and semi-automated techniques to detect and improve ambiguous RS. Following the standard guidelines for systematic mapping, review, snowballing, and quality assessment, we studied works from 2010 to 2022 on ambiguity detection and improvement techniques. We chose 42 articles as primary studies from 2,549. As a result, Natural Language Processing (NLP) and machine learning (ML) are the most promising techniques for automated and semi-automated ambiguous detection models. Furthermore, the ambiguous improvement technique began using deep learning (DL) in 2019. However, most proposed tools are still in the validation phase and are not widely employed, implying that tool development and validation research are progressing slowly. Apart from the generic linguistic context of RS, some research focuses on industrial domain-based RS. Our study shows that additional strategies have been developed to overcome RS-related issues.**

***Keywords*— Ambiguity; requirement smells; requirements specification; systematic review.**

Manuscript received 9 Jan. 2023; revised 10 May 2023; accepted 9 Sep. 2023. Date of publication 31 Dec. 2023.

I. INTRODUCTION

The software RS is the most crucial document for software quality[1]–[3]. RS outlines what software must fulfill to meet functional and non-functional needs [4]. RS contains the essential information in other software development phases, i.e., design, validation, etc. Thus, RS can impact software development project stages [5], [6]. The RS includes Business Requirements Specification (BRS), User Requirements Specification (URS), and Software Requirements Specification (SRS) [7]. Producing a quality RS document is difficult since natural language (NL) describes needs and may contain smells (e.g., ambiguity, inconsistency, etc. [8]–[10]. Critical system requirements are still expressed in ambiguous, imprecise syntax and semantics [11], [12]. Requirement smells can lead to misinterpretation, which increases the risk of time, expense overruns, and project failure [13]–[15]. Defects identified late are more expensive than those found early [16]. Ambiguity is a prevalent RS issue that can result in the deployment of a defective product [17], [18]. Ambiguous RS occurs when there are various definitions and confusion [19]. These issues may cause stakeholder confusion, unmet demands, and software products not meeting stakeholder needs [20], [21]. Ambiguity can be unacknowledged, which means that numerous readers may have various interpretations of the exact requirement if they are unaware of the ambiguity [22]. In contrast to recognized ambiguity, in which the reader is aware of the ambiguity, unacknowledged ambiguity may result in significant issues due to unconscious misinterpretation. However, manually finding ambiguous RS is tedious [22].

The Malaysian public sector has spent billions of dollars on electronic government initiatives. Much software has been developed to provide a holistic electronic government platform. Nevertheless, a semi-structured survey revealed that software developments faced delays and failures due to requirement smells (ambiguity, incompleteness etc.) [23]. Ambiguous RS was frequently mentioned as a significant contributor to software failure. NL-written RS is ambiguous, leading to various implementations later in software development [24]. According to Tukur et al. [25], programmers can interpret ambiguous requirements in their favor. These could lead to unfit systems. Common problems with software requirements, namely ambiguity, may affect software acceptance testing and subsequent phases of software development [26]. A critical quality factor is writing a straightforward SRS without ambiguity and redundancy [27]. A lengthy and fragmented statement of the requirements lowers the quality of the SRS. Therefore, a theoretical framework is formulated based on the discovered issues: "Ambiguity in Malay RS negatively affects the completion or delay of software development projects."

Motivated by the theoretical framework, our primary research goal is to devise a technique for detecting and improving ambiguous RS in the Malaysian public sector. However, developing such a technique is challenging since most available research and technologies support English and other major languages, but not the Malay RS. Thus, this paper aims to review the requirements of smell detection (focusing on ambiguity defect) and develop an automatic or semi-automatic requirement smells detection and improvement method for Malay RS. We followed [28] guidelines in conducting the systematic review. In addition, we enhance the quality of the selected final studies by applying a quality assessment [29]. The systematic review guidelines are based on the research questions (RQ) and research objective (RO) as follows:

- RQ1: What are the familiar smells in the RS studied by the researchers? (RO1: To investigate and identify the various smells in the RS.).
- RQ2: What are the common approaches for smell detection and improvement in RS? (RO2: To explore, build, and evaluate the ambiguous classification and improvement model.)

This paper contributes the following: i) explore and map various requirement smell attributes. ii) present the trend using NLP and ML in requirement smell detection. iii) Analyse the typical techniques for detecting ambiguity and improving Malay RS. iv) Present the feasibility of detecting ambiguity and improving Malay RS semi or automatically.

For the related works, Amna and Poels [30] reviewed papers from 2001 to 2020 on ambiguity requirements in user stories. According to the literature, the researchers found user stories inconsistent, insufficiently describing requirements, and duplicated functionality. Human behavior-related and cognitive elements causing ambiguity and solutions are understudied. Kaur et al. [31] surveyed published research on requirement engineering (RE) artificial intelligence (AI) techniques. This report examines 21 AI approaches for automating RE tasks. There are five (5) studies related to ambiguous requirements. Automatic classification of ML techniques outperforms manual classification procedures. Researchers employed ML classification to predict software performance from requirements. Ahmad et al. [32] examined current approaches for describing requirements for AI systems, identified available frameworks, methodologies, tools, and techniques for modeling requirements, and noted existing obstacles and limitations. The researchers conducted thorough mapping research to identify articles on current approaches to RE for AI. Ahmad et al. [32] discovered 43 research articles. The findings revealed that present RE applications were not sufficiently flexible in developing AI systems and underlined the need for new methodologies and tools to assist RE for AI. Yadav et al. [12] reviewed NLP-based RE tools for disambiguation. Study tools and processes include controlled NL, style guides, knowledge-based methods, and transfer learning. These reliable tools did not eliminate ambiguity. ML and knowledge-based produce better results.

Riaz et al. [33] reviewed 25 tools detecting the ambiguous RS between 2008 to 2018. The review is based on approaches, technologies, and ambiguities addressed. Riaz et al. [33] evaluated the tools and approach popularity using citations. RE-Context is the most cited article (196 sources). Authors ranked articles based on the number of citations. We aggregate the number of citations but select critical articles based on quality. Zhao et al. [10] conducted surveys and reviews to understand NLP in RE. The studies introduced 130 new linguistic analytic tools. Despite this, the industry has not accepted these tools, demonstrating a lack of NLP for RE standards. We found the researchers did not cover the NLP sub-techniques, i.e., tokenization etc., used in 130 NLP-based tools.

Raharjana et al. [9] conducted a systematic literature study to acquire the most recent state-of-the-art NLP research on user story-based requirements. The search approach retrieved relevant publications from six (6) trustworthy databases, such as SCOPUS, ScienceDirect, IEEE Xplore, etc. The search results are filtered using inclusion and exclusion criteria. Researchers employed both forward and reversed snowballing to generate more comprehensive results. Raharjana et al. [9] discovered 38 quality research articles that discuss NLP approaches in user stories. Researchers uncovered defects (i.e., ambiguity), developed software artifacts, identified user stories' essential abstraction, and traced model-user story relationships. Montgomery et al. [5] undertook a systematic mapping analysis to provide an overview of empirical research on requirements quality. The researchers obtained 6,905 publications from six (6) academic databases, which were reduced to 105 relevant primary papers. Empirical research on requirements quality focused on improvement strategies; few primary articles address evidence-based definitions and evaluations of quality features. Ambiguity, completeness, and consistency were the top 12 qualities that stood out the most. Researchers found 111 quality sub-types, such as "template compliance" for consistency and "passive voice" for ambiguity. Most of these subtypes contain ambiguity. Although all these studies provided rich information regarding the systematic literature review of requirement specifications' quality and defects, none focused on ambiguity detection and improvement for Malay RS.

The rest of this article is assembled as follows. Section II reports the approach employed to conduct the research. This section consists of three stages, i.e., planning, conducting, and documenting the review. Following this, section III reports the results and discusses the findings. Finally, the conclusion is presented in section IV.

II. MATERIALS AND METHOD

We followed a thorough and reliable series of methodological Budgen and Brereton [28] guidelines to study literature. The research methodology overview is illustrated in Fig. 1.

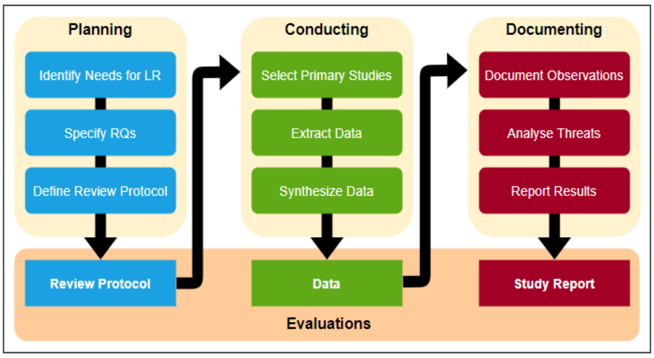

Fig. 1 Overview of research methodology

A. Planning the Review

The need for a systematic review leads to the following study strategy: Step 1-Recognise the need for review. Section I contains the RQ that guides the systematic review methodology creation and evaluation. Step 2-Identify the RQs. The RQs provide the basis for developing a search strategy for extracting literature. The rationale for each question defines the fundamental goal of the investigation. Step 3 - Define and assess the review procedure. We describe the RQ and scope to build search strings for literature extraction.

B. Conducting the Review

Starting with study selection and ending with extracted data and synthesized information, the second phase is conducting the review: Step 1-Select primary studies. Budgen and Brereton [28] guidelines generated the search phrases based on the RQs. Fig. 2 shows the composition of 5,544 search strings applied to seven (7) credible databases, including ACM, IEEE Xplore, Science Direct, Scopus, Springer Link, Wiley, and Google Scholar. We gathered 2,549 peer-reviewed articles from 2010 to 2022.

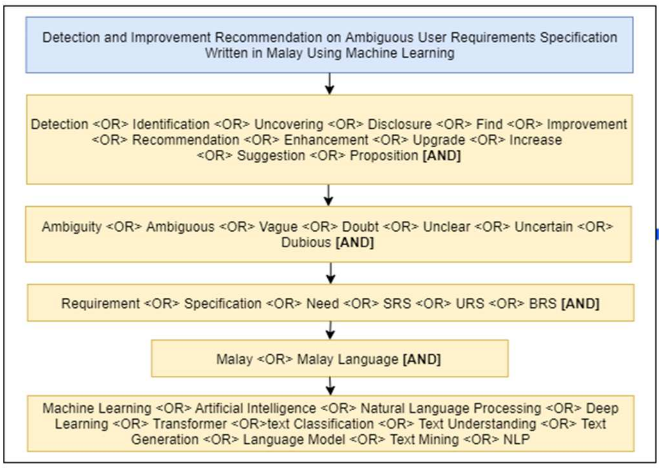

Fig. 2 Composition of search strings and search results

The primary study's selection processes include database search, inclusion/ exclusion, snowballing, and final selection with quality assessment, as illustrated in Fig. 3.

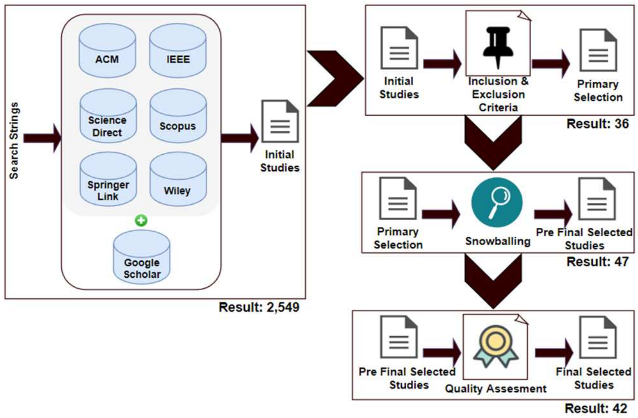

Fig. 3 Primary study selection process

Initial Selection: The researchers examine the titles of possible primary studies against the inclusion/ exclusion criteria in Table I.

TABLE I
INCLUSION AND EXCLUSION CRITERIA

	Criteria	Rationale
Inclusion (I)	I1: The article is within the context of research.	A research article ensures peer review consistency and has much material.
	I2: The article presents solutions, knowledge, or measurements for identifying and recommending ambiguous RS.	Ambiguity detection and improvement RS require concrete solutions, metrics, and evaluations.
Exclusion (E)	E1: The articles do not address ambiguity RS detection and improvement.	We aim to study the detection and improvement recommendation of ambiguity RS, excluding any other ambiguity.
	E2: The articles do not suggest a process, method, or tool to detect and recommend ambiguous RS.	These studies do not explicitly detect or even recommend ambiguous RS.
	E3: The articles do not include editorials, abstracts, or brief articles (less than six (6) pages).	These studies do not provide unbiased knowledge.
	E4: The articles do not include secondary studies.	These studies propose no approach.
	E5: The articles do not include non-peer-reviewed studies, white papers, manuscripts not in English and Malay (excluding articles not in English and Malay), or unbelievable sources.	Since ambiguity requirements are too context-specific for other domains, we decided to exclude them.
	E6: The articles do not include magazine or non-academic articles.	These studies do not provide non-academic or research content.

Final Selection: After the inclusion/ exclusion task, 36 studies were chosen. Snowballing from identified article reference lists should be employed alongside database searches to find more relevant articles. This method resulted in the inclusion of seven (7) meaningful studies. Thus, 47 studies were selected for qualitative assessment. Qualitative Assessment of Included Studies: We ranked these studies using numerical quality ratings. The number of citations did not represent the articles' quality. We followed the qualitative assessment guidelines by Jamshidi et al. [29]. Table II shows the quality assessment checklist.

TABLE II
CHECKLIST FOR QUALITY ASSESSMENT

General Items for Quality Assessment (A)	**Score** Yes = 1	Partially = 0.5	No = 0
A1: Are the definition of the issue and the study's motivation presented?			
A2: Is the research setting clarified in which the analysis was performed?			
A3: Is the methodology of the study and its organization stated clearly?			
A4: Are the research contributions in line with the findings presented?			
A5: Are the observations and lessons learned directly stated from the research?			
Specific Items for Quality Assessment (B)			
B1: Is the study based on ambiguity requirements detection and improvement recommendations?			
B2: Are the specifics of relevant research specifically discussing the identification and recommendation of ambiguity requirements?			
B3: Does the research assessment explain the research methodology clearly?			
B4: Are the outcomes in a non-trivial assessment sense explicitly validated?			
B5: Are drawbacks and potential repercussions for the identification and improvement recommendation of ambiguity requirements positioned?			

We established a quality ranking based on the quality assessment checklist. A1 to A5 reflects the general assessment criteria, with a 25% maximum score each; meanwhile, B1 to B5 reflect the specific assessment criteria, with a 75% maximum score each. The highest score for the qualitative assessment is 4. i) Score 3 to 4: quality, ii) Score 1.5 to 2.99: acceptable, iii) Score below 1.5: eliminate. The quality score formula is as follows:

$$Quality\ Score\ = \left[\frac{\sum_{i=1}^{5} Ai}{5} + \left(\frac{\sum_{i=1}^{5} Bi}{5}\ x\ 3\right)\right] \quad (1)$$

Ai = General Items for Quality Assessment (A)
Bi = Specific Items for Quality Assessment (B)

We selected 42 articles as primary studies with bibliographies, quality scores, and citations. The quality ranking was an internal statistic that helped us choose the most relevant studies; it did not reflect any comparison or external assessment. Step 2 - Data extraction. In this step, we created data extraction forms that accurately recorded the information collected from primary studies. Data extraction forms should be created and piloted once the study protocol is determined to minimize the risk of bias. Step 3 - Data synthesis. Data synthesis required compiling and including a summary of key research findings.

C. *Documenting the Review*

The final phase of a systematic review entails writing up based on synthesized data. We answered the RQs based on the selected studies' retrieved data, reported the results, and discussed the insights in Section III.

III. Results and Discussion

This section explains the findings of this study and answers the RQ stated in Section I. Also, this section discusses the findings. Also, we discuss the threat to validity.

A. *Overview of the Primary Studies*

Table III indicates that most articles were about ambiguous English RS, but less for Malay. As a result, 2,549 articles were listed for initial studies. The number of articles was reduced to 36 after primary selection.

TABLE III
PRIMARY STUDY SELECTION RESULT

Database	Initial studies	Primary selection	Snowballing	Quality Assessment
ACM	402	8	7	5
IEEE	19	7	12	12
Science Direct	323	1	2	2
Scopus	261	12	13	13
Springer Link	1,511	6	8	5
Wiley	31	0	0	0
Google Scholar	2	2	5	5
Total:	**2,549**	**36**	**47**	**42**

Next, the snowballing task was performed to get the related articles. The total number of articles after the snowballing task was 47. The final total of selected articles through the quality assessment task was 42. The most contributed database was Scopus. The primary study selection result by phases is illustrated in Fig. 4.

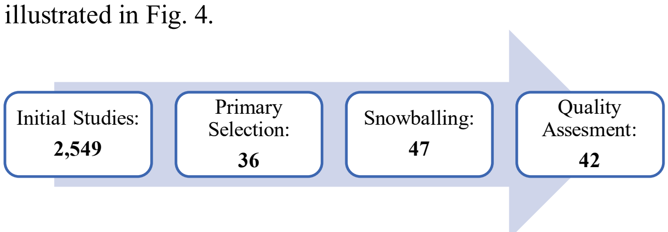

Fig. 4 Primary study selection process result by phases

B. *Systematic Mapping on Distribution Studies (RQ1)*

Fig. 5 shows a systematic mapping [34] of the distribution studies according to requirement smells attributes, contribution category, and research category answering RQ1 in Section I. We found ambiguity, inconsistency, incompleteness, and other requirements smell. The ambiguity associated with the method, process, and solution proposal was 28 frequencies. There were 26 ambiguities related to evaluation research. No opinion paper was associated with any of the defect attributes.

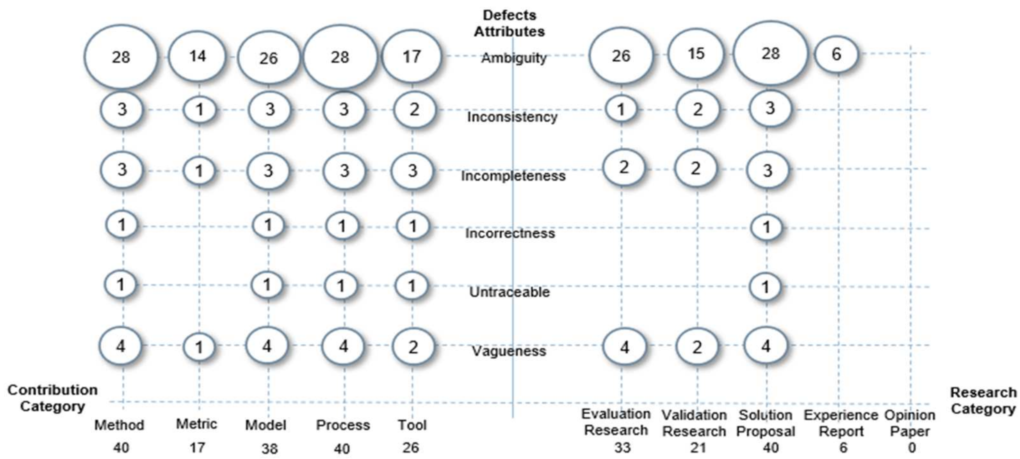

Fig. 5 Systematic mapping of distribution studies

C. *Distribution of Studies According to Typical Approaches (RQ2)*

NLP was commonly employed to determine ambiguous RS. The NLP consists of word vectors, tokenization, n-gram, etc. ML, especially the classification algorithm, was the second most preferred technique. The third common technique was elicitation. Fig. 6 illustrates study techniques. Fig. 7 shows trends of ambiguous detection and improvement for RS using NLP techniques. In 2017, we found tokenization, semantic similarity, word-to-vector, part-of-speech, parsing techniques, and the Stanford CoreNLP toolkit popular. ML classification and rules-based algorithms increased in 2011 but declined in 2012. The same pattern repeated in 2018 but changed in 2019. Naive Bayes, Random Forest, and Rules-based were the most popular techniques. Fig. 8 shows the ML and rules-based approaches trend.

Fig. 9 illustrates the common conceptual framework for ambiguity requirement detection and improvement model: Input, Process, and Output. The RS was an input for the process. The unambiguous requirement was an output. The process has two models: ambiguity detection and ambiguity improvement model. The ambiguity detection model consists of NLP and ML/ Rule-based; meanwhile, the ambiguity improvement model disambiguates ambiguity terms and generates unambiguous RS candidates. Some toolkits and techniques did not support Malay, but some supported, such as Generative Pre-Trained (GPT) (DL) and BabelNet.

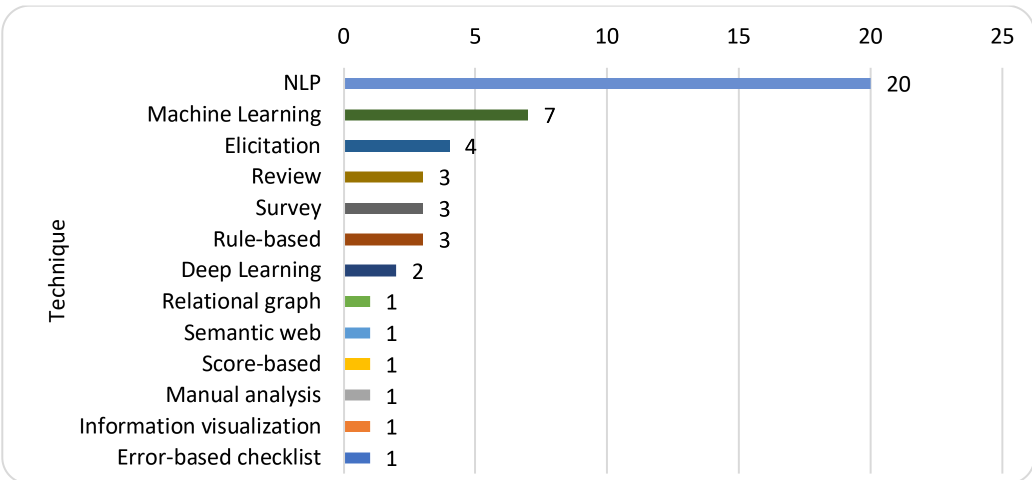

Fig. 6 Distribution of studies according to techniques

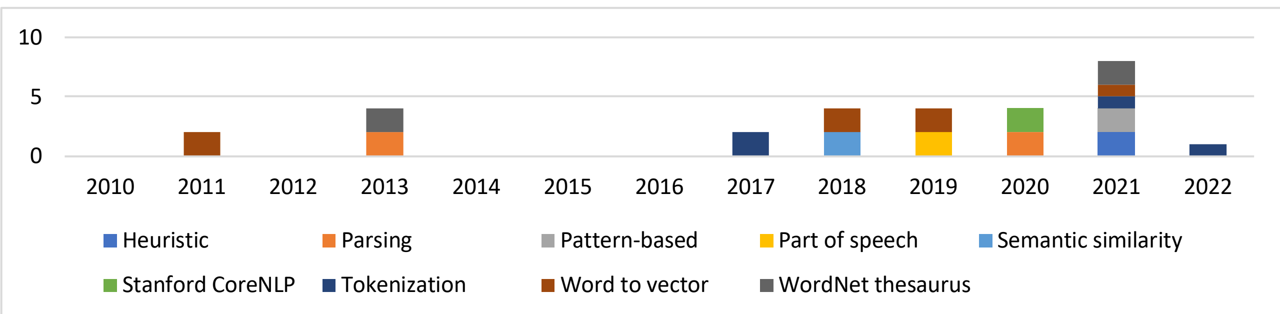

Fig. 7 Trends in NLP techniques from 2010 to 2022

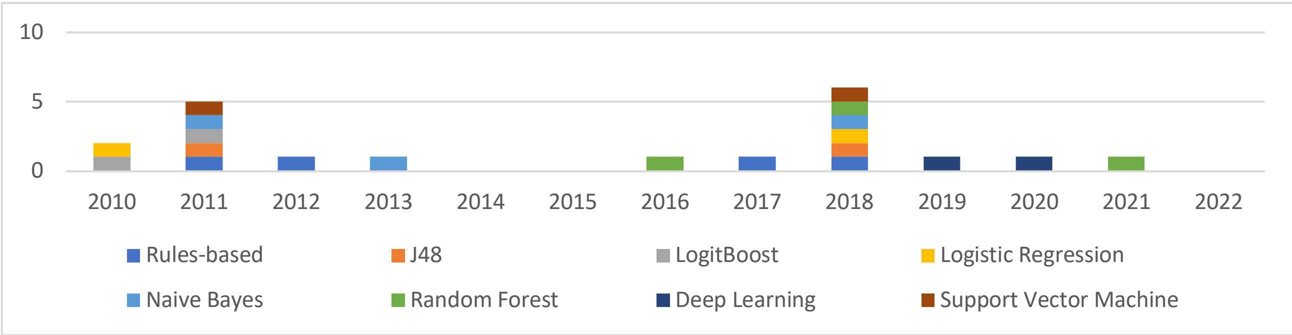

Fig. 8 Trends in ML and rules-based techniques from 2010 to 2022

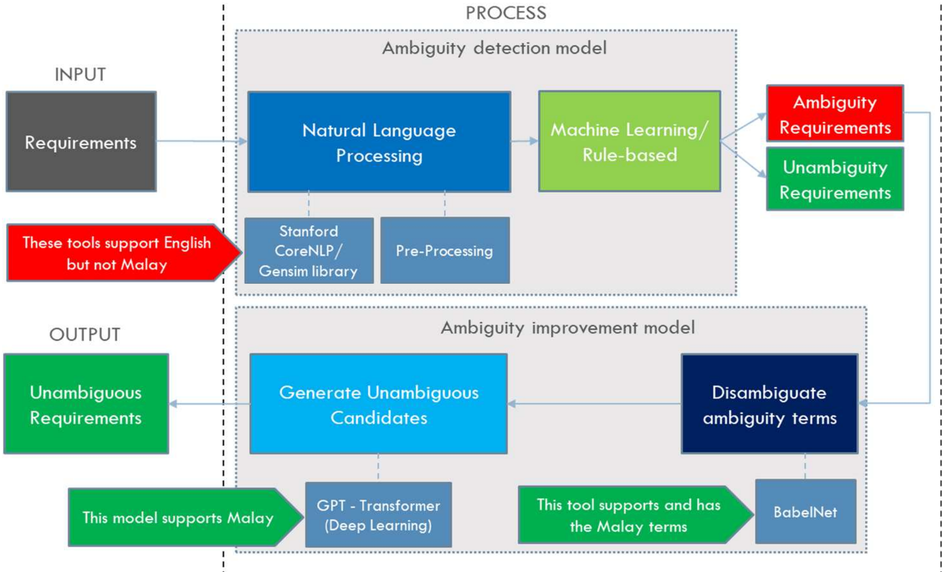

Fig. 9 Common conceptual framework based on past research

D. *Research Trends and Future Directions*

1) *Current techniques and expected future path:* Several techniques for detecting ambiguity in RS were discovered. NLP is the most common method for determining ambiguous RS. NLP techniques can be applied to Malay RS, i.e., tokenization, word-to-vector, and part-of-speech. There are NLP toolkits for English and other primary languages. Aside from that, DL techniques started being employed in NLP tasks in 2019. We believe DL will improve the ambiguous RS and support other languages, including Malay. After NLP, ML and Rules-based techniques are widely used. Previously, Naive Bayes and Random Forest were the most popular classification algorithms. These techniques are based on the suitability and features of the collected data. From 2011 to 2017, these methods were employed but with less frequency. Also, in 2018, these methods inclined. From 2012 to 2017, we assumed these techniques were used little. However, ML and rule-based systems will take time to evolve in this field.

2) *The need for a comprehensive ambiguity improvement model:* We discovered some unresolved research issues that could lead to new research directions. We found that most previous studies focus on ambiguity detection. However, less research has been done on improving RS. We hope the DL technique will mature for the ambiguity improvement model through NL generation.

E. *Application Domains Affect Model Performance*

Ferrari et al. [35] identified and ranked ambiguous terms across five (5) domains. The word2vec algorithm uses domain-specific documents to learn word embeddings from the corpus. The method works well with a few domains but not well with many. According to Ferrari et al. [36], NLP and rule-based smell detection are applied to railway industry requirements. The authors' approach is unsuited to other RS domains. However, the author's model specificity domains could be fine-tuned to improve performance.

F. *The Growing Domains-Based RS*

Most previous research focused on the RS's generic linguistic context. Several articles focused on industry domains like railway, medical, and engineering. These requirements are domain-based and are slowly growing. Research on cross-domain ambiguous terms has focused on possible words found in more than one domain with multiple meanings. However, if the model covered multiple domains, the model's performance decreased.

G. *Language Models are Complex but Good in-NL Generation*

GPT is a pre-trained language model based on DL. GPT-2 and GPT-3 can learn a word sequence's likelihood and predict the next word. These models were trained on 19 billion WebText2 tokens [37][38]. GPT-3 optimizes complex language models to improve task-agnostic performance and could enhance ambiguous RS.

H. Threats to Validity

This review categorizes and compares research on ambiguous RS detection and improvement. While systematic reviews [28] are often credible, some drawbacks exist. While this method reduces bias, it increases search time. A review protocol was created to define relevant articles and ensure fair selection. Data extraction and selection may have consistency issues. Each systematic review study is evaluated for quality:

1) Internal validity: Fig. 1 describes the systematic review search strategy [28]. A few articles for Malay requirements were in the primary search results from credible databases. Due to the broad knowledge, an additional search was chosen to employ Google Scholar to find articles related to the ambiguous Malay RS.

2) Construct validity: Fig. 3 shows the snowballing task collected 47 articles. The construct validity threatens the articles' quality. The number of citations did not represent the quality of the article. To minimize risk, we employed qualitative assessment [29].

IV. Conclusion

This study reviewed prior work on ambiguous Malay RS detection and improvement. NLP and ML/ Rule-based techniques are commonly employed to identify ambiguous smells in RS. An ambiguity detection model and an ambiguity improvement model were presented. Most tools support English and other major languages. However, some tools support Malay RS. Most validated tools are not for industry-wide use. This implies a gap between research and practice. Despite the low research on improving ambiguous RS, we expect the DL and related approaches to mature via the NL generation strategy. These methods may evolve for future research.

Acknowledgment

Malaysia's Ministry of Higher Education funded this research under the Fundamental Research Grant Scheme (FRGS/1/2020/ICT01/UPM/02/1). The Federal Training Award (HLP) 2020 from the Public Service Department of Malaysia supported this effort.

References

[1] A. Spillner and T. Linz, *Software Testing Foundations: A Study Guide for the Certified Tester Exam- Foundation Level- ISTQB® Compliant*. dpunkt.verlag, 2021.

[2] A. Belfadel, J. Laval, C. Bonner Cherifi, and N. Moalla, "Requirements engineering and enterprise architecture-based software discovery and reuse," *Innov. Syst. Softw. Eng.*, vol. 18, no. 1, pp. 39–60, 2022, doi: 10.1007/s11334-021-00423-5.

[3] M. A. Jubair *et al.*, "A multi-agent K-means with case-based reasoning for an automated quality assessment of software requirement specification," *IET Commun.*, 2022, doi: 10.1049/cmu2.12555.

[4] S. F. Alshareef, A. M. Maatuk, T. M. Abdelaziz, and M. Hagal, "Validation framework for aspectual requirements engineering (ValFAR)," 2020, doi: 10.1145/3410352.3410777.

[5] L. Montgomery, D. Fucci, A. Bouraffa, L. Scholz, and W. Maalej, "Empirical research on requirements quality: a systematic mapping study," *Requir. Eng.*, vol. 27, no. 2, pp. 183–209, 2022, doi:10.1007/s00766-021-00367-z.

[6] M. A. Akbar, A. Alsanad, S. Mahmood, A. A. Alsanad, and A. Gumaei, "A Systematic Study to Improve the Requirements Engineering Process in the Domain of Global Software Development," *IEEE Access*, vol. 8, pp. 53374–53393, 2020, doi:10.1109/access.2020.2979468.

[7] E. D. Canedo and B. C. Mendes, "Software requirements classification using machine learning algorithms," *Entropy*, vol. 22, no. 9, Sep. 2020, doi: 10.3390/E22091057.

[8] I. García, C. Pacheco, A. León, and J. A. Calvo-Manzano, "A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148 systems and software engineering – Lifecycle processes – Requirements engineering at undergraduate level," *Comput. Stand. Interfaces*, vol. 67, p. 103377, 2020, doi: https://doi.org/10.1016/j.csi.2019.103377.

[9] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Stories and Natural Language Processing: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 53811–53826, 2021, doi: 10.1109/ACCESS.2021.3070606.

[10] L. Zhao *et al.*, "Natural Language Processing for Requirements Engineering," *ACM Comput. Surv.*, vol. 54, no. 3, Apr. 2021, doi:10.1145/3444689.

[11] M. Osama, A. Zaki-Ismail, M. Abdelrazek, J. Grundy, and A. Ibrahim, "A Comprehensive Requirement Capturing Model Enabling the Automated Formalisation of NL Requirements," *SN Comput. Sci.*, vol. 4, no. 1, p. 57, 2022, doi: 10.1007/s42979-022-01449-7.

[12] A. Yadav, A. Patel, and M. Shah, "A comprehensive review on resolving ambiguities in natural language processing," *AI Open*, vol. 2, pp. 85–92, 2021, doi: 10.1016/j.aiopen.2021.05.001.

[13] A. Hussain, H. Ahmed, A. Khamaj, and M. N. M. Nawi, "a Model of Consequences of Ambiguous Requirements," *J. Southwest Jiaotong Univ.*, vol. 56, no. 6, pp. 599–609, 2021, doi: 10.35741/issn.0258-2724.56.6.52.

[14] C. Ribeiro and D. Berry, "The prevalence and severity of persistent ambiguity in software requirements specifications: Is a special effort needed to find them?," *Sci. Comput. Program.*, vol. 195, p. 102472, 2020, doi: 10.1016/j.scico.2020.102472.

[15] A. Fantechi, S. Gnesi, and L. Semini, "VIBE: Looking for Variability In ambiguous requirements," *J. Syst. Softw.*, vol. 195, p. 111540, 2023, doi: 10.1016/j.jss.2022.111540.

[16] J. Iqbal, R. B. Ahmad, M. Khan, M. H. Nizam, and A. Akhunzada, "Model to Cope with Requirements Engineering Issues for Software Development Outsourcing," *IEEE Access*, vol. 10, pp. 63199–63229, 2022, doi: 10.1109/ACCESS.2022.3182393.

[17] M. R. Asadabadi, E. Chang, O. Zwikael, M. Saberi, and K. Sharpe, "Hidden fuzzy information: Requirement specification and measurement of project provider performance using the best worst method," *Fuzzy Sets Syst.*, vol. 383, pp. 127–145, 2020, doi:10.1016/j.fss.2019.06.017.

[18] K. H. Oo, "Comparing Accuracy Between SVM, Random Forest, K-NN Text Classifier Algorithms for Detecting Syntactic Ambiguity in Software Requirements," in *Lecture Notes in Networks and Systems*, 2023, vol. 550 LNNS, pp. 43–58, doi: 10.1007/978-3-031-16865-9_4.

[19] A. Griva, S. Byrne, D. Dennehy, and K. Conboy, "Software Requirements Quality: Using Analytics to Challenge Assumptions at Intel," *IEEE Softw.*, vol. 39, no. 2, pp. 80–88, 2022, doi:10.1109/MS.2020.3043868.

[20] S. Ezzini, S. Abualhaija, C. Arora, and M. Sabetzadeh, "Automated Handling of Anaphoric Ambiguity in Requirements: A Multi-Solution Study," in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 187–199, doi:10.1145/3510003.3510157.

[21] F. Dalpiaz, I. van der Schalk, S. Brinkkemper, F. B. Aydemir, and G. Lucassen, "Detecting terminological ambiguity in user stories: Tool and experimentation," *Inf. Softw. Technol.*, vol. 110, pp. 3–16, 2019, doi: 10.1016/j.infsof.2018.12.007.

[22] S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, and L. C. Briand, "Using domain-specific corpora for improved handling of ambiguity in requirements," in *Proceedings - International Conference on Software Engineering*, May 2021, pp. 1485–1497, doi:10.1109/ICSE43902.2021.00133.

[23] M. F. Zahrin, M. H. Osman, A. A. Halin, S. Hassan, and A. Haron, "Issues in Requirements Specification in Malaysia" s Public Sector: An Evidence from a Semi-Structured Survey and a Static Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 11, pp. 284–292, 2022, doi:10.14569/IJACSA.2022.0131132.

[24] F. Ashfaq and I. S. Bajwa, "Natural language ambiguity resolution by intelligent semantic annotation of software requirements," *Autom. Softw. Eng.*, vol. 28, no. 2, Nov. 2021, doi: 10.1007/s10515-021-00291-0.

[25] M. Tukur, S. Umar, and J. Hassine, "Requirement Engineering Challenges: A Systematic Mapping Study on the Academic and the Industrial Perspective," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3723–3748, 2021, doi: 10.1007/s13369-020-05159-1.

[26] O. M. H. Et.al, "Ambi Detect: An Ambiguous Software Requirements Specification Detection Tool," 2021. doi:10.17762/turcomat.v12i3.1066.

[27] J. Medeiros, A. Vasconcelos, C. Silva, and M. Goulão, "Requirements specification for developers in agile projects: Evaluation by two industrial case studies," *Inf. Softw. Technol.*, vol. 117, p. 106194, Jan. 2020, doi: 10.1016/j.infsof.2019.106194.

[28] D. Budgen and P. Brereton, "Performing systematic literature reviews in software engineering," *Proc. - Int. Conf. Softw. Eng.*, vol. 2006, pp. 1051–1052, Aug. 2006, doi: 10.1145/1134285.1134500.

[29] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 142–157, 2013, doi: 10.1109/TCC.2013.10.

[30] A. R. Amna and G. Poels, "Ambiguity in user stories: A systematic literature review," *Inf. Softw. Technol.*, vol. 145, p. 106824, 2022, doi:10.1016/j.infsof.2022.106824.

[31] K. Kaur, P. Singh, and P. Kaur, "A review of artificial intelligence techniques for requirement engineering," in *Advances in Intelligent Systems and Computing*, 2021, vol. 1257, pp. 259–278, doi:10.1007/978-981-15-7907-3_20.

[32] K. Ahmad, M. Abdelrazek, C. Arora, M. Bano, and J. Grundy, "Requirements engineering for artificial intelligence systems: A systematic mapping study," *Inf. Softw. Technol.*, vol. 158, 2023, doi:10.1016/j.infsof.2023.107176.

[33] M. Q. Riaz, W. H. Butt, and S. Rehman, "Automatic Detection of Ambiguous Software Requirements: An Insight," in *5th International Conference on Information Management, ICIM 2019*, 2019, pp. 1–6, doi: 10.1109/INFOMAN.2019.8714682.

[34] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," 2008, doi:10.14236/ewic/ease2008.8.

[35] A. Ferrari and A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering," *Autom. Softw. Eng.*, 2019, doi: 10.1007/s10515-019-00261-7.

[36] A. Ferrari *et al.*, "Detecting requirements defects with NLP patterns: an industrial experience in the railway domain," *Empir. Softw. Eng.*, vol. 23, no. 6, pp. 3684–3733, 2018, doi: 10.1007/s10664-018-9596-7.

[37] L. Reynolds and K. McDonell, "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm," 2021, doi:10.1145/3411763.3451760.

[38] X. V. Lin *et al.*, "Few-shot Learning with Multilingual Generative Language Models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Dec. 2022, pp. 9019–9052, doi: 10.18653/v1/2022.emnlp-main.616.