Systematic Literature Review Langchain Proposed

1st Rakha Asyrofi
Faculty of Marine Science and Engineering
Universitas Hangtuah
Surabaya, Indonesia
Department of Computer Science and Information Engineering
National Central University
Taiwan
asyrofi@hangtuah.ac.id

3rd Muhammad Irfan Lutfhi
Faculty of Engineering
Universitas Negeri Yogyakarta
Yogyakarta, Indonesia
Graduate Institute of Network Learning Technology
National Central University
Taiwan
m.irfanluthfi@uny.ac.id

2nd Mutia Rahmi Dewi*
Department of Information Technology
Politeknik Negeri Padang
Padang, Indonesia
mutiarahmi@pnp.ac.id

4th Prasetyo Wibowo
Department of Informatics and Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
pras@pens.ac.id

2023 International Electronics Symposium (IES) | 979-8-3503-1473-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/IES59143.2023.10242497

***Abstract*—While systematic literature reviews are frequently carried out within software engineering research, performing them in a rigorous and reproducible manner can be difficult. This paper proposes some new methods for evaluating and validating systematic literature reviews. Our approach consists of several steps, such as: Selecting a set of relevant scientific papers to analyze, Developing a list of questions and criteria to evaluate each literature review, and Determining what types of functionality and performance should be evaluated. We tested our method by having multiple experts evaluate the literature reviews based on our questions and criteria. We measured the similarity in scores between each expert to determine the reliability of the evaluations. The average similarity index between experts was 0.58 to 0.83, indicating a reasonable level of agreement in their assessments. This shows our evaluation method can produce fairly consistent results, even when different experts are involved. The relatively high level of agreement is notable considering each expert brings their perspectives and opinions in analyzing literature reviews. By providing concrete questions, criteria, and evaluation methods, we aimed to guide the experts toward more uniform evaluations. In summary, we developed and tested a new approach for evaluating and validating systematic literature reviews in software engineering. By assessing reliability via inter-rater agreement, we showed that consistent and reproducible results are possible using our evaluation framework and methodology. Our methods could help researchers gain more insight into what makes for an effective and high-quality literature review.**

***Index Terms*—Documents Review, LangChain, Systematic Literature Review, Software Engineering**

I. Introduction

A systematic Literature Review (SLR) is a rigorous and structured process to review and summarize existing research on a specific topic or issue. In contrast to other types of literature reviews, which may be less systematic or complete, SLRs adhere to established protocols and try to eliminate bias by employing explicit and transparent methodologies. There are various drawbacks to conducting manual SLRs that can reduce their usefulness and efficiency. One of the main disadvantages is the large time and resource investment required. Manually searching, retrieving data, and synthesizing findings can take time for researchers and review teams, thereby delaying study progress [1].

Manual reviews present extra issues due to subjectivity and bias. Subjective judgments and biases of researchers can influence study selection, data extraction, and interpretation processes, and trustworthiness of the review [2]. This issue can introduce inaccuracies and undermine the legitimacy of the review conclusions.

Another disadvantage of manual systematic reviews is incomplete and biased coverage of the literature. Manual searching may miss important studies due to limits in search tactics, database restrictions, and language constraints. As a result, the review may have suffered from inadequate coverage of the existing literature, which may have resulted in biased results. [3]

Manual systematic reviews have similar reproducibility and transparency issues. A lack of adequate reporting on the review process, including searching techniques, studying selection criteria, and data extraction procedures, might impede reproducibility and make updating or replicating the review difficult, [4]. Inadequate reporting reduces the review's transparency, making it difficult for outsiders to assess its rigor and correctness.

As a result, several studies have discovered that automated systematic literature reviews outperform manual methods in several ways. According to Marshall [5], automated procedures save time and effort for researchers and review teams. By employing algorithmic methods and machine learning to automate review activities such as literature search, data extraction, and analysis, this automation can boost productivity and assist academics in better managing their resources. Furthermore, automated systematic literature reviews, according to Maraeves et al. [6], can improve the efficiency and accuracy of

the review process. It can swiftly discover relevant research, extract essential information, and evaluate enormous amounts of data using advanced computational techniques such as text mining and machine learning, ensuring a more rigorous and exact synthesis based on available evidence. This strategy can reduce the danger of human mistake and bias that comes with manual review.

Furthermore, automated systematic literature can handle the growing number of scientific research. The increased volume allows for better scope and scalability and the rapid examination of vast amounts of literature from numerous data sources. Compared to the manual procedure, this strategy enables a more thorough analysis and ensures a broader scope of the research process [7]. Another benefit is that the technology automates examining and confirming data, allowing other researchers to assess and replicate the findings quickly. The SLR update method can ensure the relevance of trustworthy data returns regularly [8].

Based on the problem's discussion, an approach consists of several significant components. In brief, the primary contributions of our study are:

1) Modeling based on multiple scenarios and comparing to new approaches utilizing LangChain to produce automated systematic literature reviews.
2) Testing multiple scenarios and tests based on selecting scientific articles as source documents, evaluation questions, and functionality testing.
3) Collect expert assessments, development processes, functionality, and performance testing results.

II. Related Work

In recent years, there are several studies and research about literature review that have been conducted in the Natural Language Processing (NLP) research area, including system demonstrations that showcase the potential and advancements of NLP techniques. This paper explores and examines the latest techniques for gaining insights from text data while highlighting the key advantages and disadvantages of each method [9].

Alquliti et al. [10] introduce the use of word2vec to build matrix representations for summarizing individual documents. The convolutional neural network (CNN) serves as the foundation and training and testing were conducted on a dataset of 100 documents from the DUC2002 dataset. They explored 26 different CNN configurations to identify the optimal architecture for predicting informative sentences. They found that shallower representation depths corresponded to better selection accuracy.

Another study discusses how people's trust in the accuracy of online news articles can impact their subsequent actions and choices. The perceived credibility of digital news sources plays an important role in determining what information a person believes and how they respond to that information [9]. and other research has investigated how traditional literature review methods can be enhanced using natural language processing techniques, such as automated information retrieval. These studies demonstrate how NLP can augment and strengthen the usual process of bibliographic literature analysis [11].

Some experts in information science make use of selected sample documents, known as "seed studies," before crafting search queries. By testing queries against these seed studies first, they can evaluate how effective the queries are at retrieving relevant information. This helps ensure the search queries are optimized before conducting a full-scale search of the literature. [12].

Other research aims to gain insights into how users interact with and evaluate information retrieval systems. Understanding user behavior and preferences is crucial for developing effective interactive information retrieval (IIR) technologies. By studying how people use and judge IR systems, researchers can build systems that better align with user requirements and expectations. [13]. and then another resource proposes a new query-by-document method that uses Monte Carlo sampling of important keywords. Given a set of input documents (the seeds), keywords are extracted using the term frequency-inverse document frequency (TF-IDF). These keywords are then randomly sampled to iteratively generate search queries, which are applied to a database. [14].

Another paper reveals intuitions and behaviors by analyzing the query logs of a specialized tool developed to assist expert searchers in refining complex boolean queries [15]. While there are many ethical guidelines for AI systems in general, these principles are often difficult to apply to specific AI use cases, such as employing AI algorithms for recruitment and hiring. [16]. While numerous guides on literature reviews exist, these are often limited to the philosophy of review procedure, protocols, and nomenclature, triggering, non-parsimonious reporting and confusion due to overlapping similarities [17],

Some research seeks to systematically analyze the existing literature on using automated tools for web accessibility evaluation [18], Their research finds that, aside from some work on crowdsourcing relevance ratings, few studies have explored the use of gamification techniques for search systems [19].

CS-SMATS Algorithm was conducted by Patil et al. which utilizes the cuckoo search optimization algorithm, to generate concise summaries from input documents while preserving their main content. This model aims to save users' time by providing a brief and compressed version of the document, allowing for easier data retrieval and helping users determine if a document is worth reading. This algorithm can summarize both single and multiple documents, ensuring readability and high-quality language in the generated summaries, ensuring that the summary covers all important aspects of the original document's content, and voiding redundancy in the generated summaries [20].

III. Methodology

Figure 1 illustrates the methodology that consists of several steps, there are Research and Development, then a selection of scientific articles as document sources conducts an expert evaluation process and carries out a development process and tests in the form of functionality tests and performance

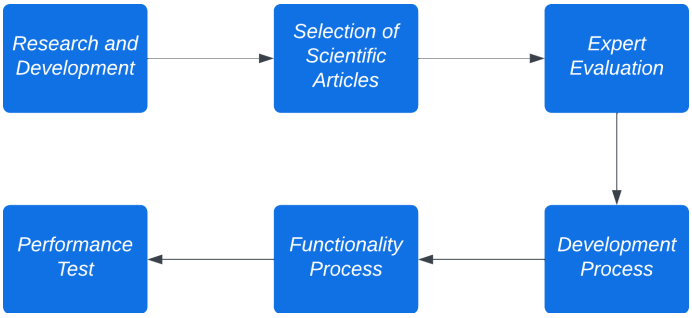

Fig. 1. Methodology

tests. The more detailed explanation of how this methodology works, it will be explained below:

1) **Research and Development**: This section explains the research and development process that was used in this study. The method is carried out to discover related research linked to SLRs and the current state of the art in order to obtain a system that can solve current difficulties.
2) **Selection of Scientific Articles**: A variety of publications will be used as references for the scenarios that will be performed. The experimental scenario includes five primary articles with published criteria and the complexity of the written content. The complexity of the system is crucial since it is used to test its reliability.
3) **Expert Evaluation**: The evaluation of the Subject Matter Expert (SME) is crucial in this study since it can lessen the findings of the system's bias. Six SMEs will examine each article provided based on their different research disciplines, with the SME results serving as the primary standard for the system developed.
4) **Development Process**: At this point, work on the system will begin. This advancement is based on the findings of research conducted to meet the research's objectives. We aim to test the amount of reliability of the model we construct using two situations.
5) **Functionality Test**: The system's functionality will be examined to ensure that it operates correctly and effectively. Process input, document conversion, indexing, and tokenization are among the functions performed.
6) **Performance Test**: The final element in this research technique is performance testing. The res performance will assess the model's dependability and results, allowing the suggested model to be compared

IV. Experiments

A. Baselines

We conducted a comparison of state-of-the-art models based on different scenarios. The scenarios are as follows:

1) **First Scenario**: We utilized several tools and libraries starting from LangChain, Embedding API, and LLM Generative AI as shown in Figure 2
2) **Second Scenario**: We build a semantic index Knowledge Base using FAISS, which ranked the system response according to answer criteria as shown in Figure 3
3) **Our Approach**: We aimed to create embedding API cooperation, specifically using OpenAI's embedding and Huggingface Instructor Embeddings. We stored and processed the built data based on the semantic index as a knowledge base using FAISS, utilizing the Large Language Model (LLM) Generative AI, Specifically the Davinici-Model as shown in Figure 4

To verify the effectiveness of each component, we conducted the following ablation experiments:

1) **Diff. LLMs**: In this case, we compared the results when we are using different LLMs. However, we found that the impact was not significant enough
2) **Diff. vector stores**: By comparing different vector store methods for storing the knowledge base data, the system facade difficulties in providing clear explanations and generating arguments. Hence, the results were not significantly different.

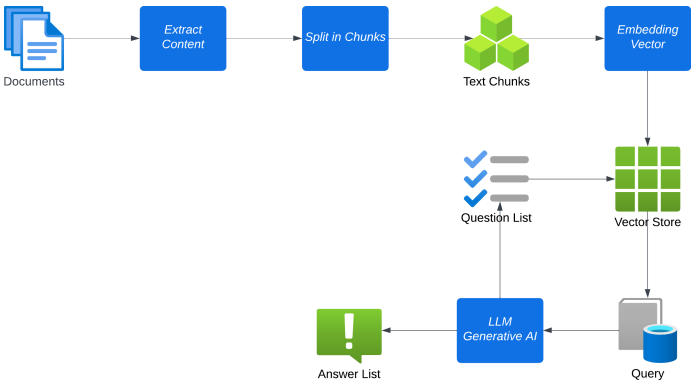

Fig. 2. First Scenario

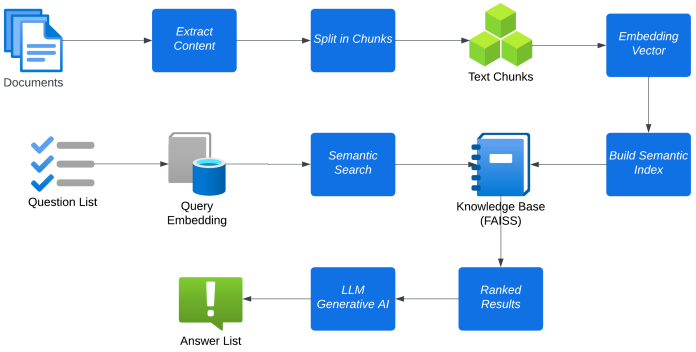

Fig. 3. Second Scenario

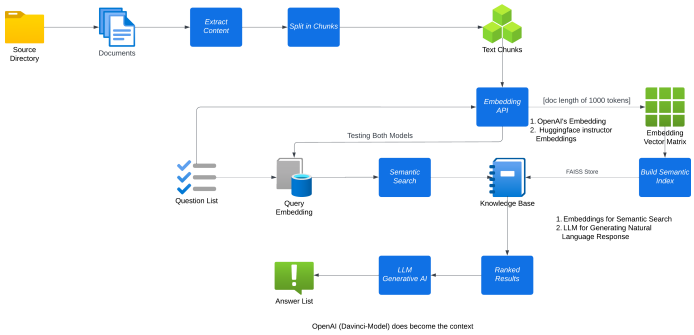

Fig. 4. Our Proposed

TABLE I
ARTICLE & QUESTION INDEX EVALUATION SCORE

Art/ Q.	Q1	Q2	Q3	Q4	Q5	Q6
Art1.	0.65	0.68	0.67	0.65	0.83	0.65
Art2.	0.65	0.77	0.67	0.68	0.83	0.65
Art3.	0.65	0.69	0.67	0.69	0.66	0.65
Art4.	0.65	0.61	0.66	0.65	0.65	0.59
Art5.	0.64	0.62	0.63	0.60	0.60	0.56

B. Implementation Details

We implement all models in Python 3.10.7 with the environment jupyter notebook or Python console and do some importing libraries. That includes document loaders by PyPDFloader and directoryloader, text to index by LaMini-LM [21] and OpenAI llms, embedding model by OpenAIEmbeddings and InstructorEmbeddings [22], vector stores by FAISS [23], and also answer+reasoning model by GPT-3 Text-dalvik-003 Model.

We want to compare all that performance based on several hyper-parameter embedding that consists of a temperature parameter is 0.1, max length parameter is 100, chunk size parameter is 1000, chunk overlap parameter is 200, pad token id is 50256, top p is 0.95, repetition penalty is 1.15, k search kwargs is 3. We use the instructorEmbeddings to achieve the best performance.

V. EVALUATION METRICS

A. Expert Evaluation

We have done several performances based on a calculation using the string similarity method [24], evaluate the relevance of module answer evaluation to user questions based on our expert answer, assess the accuracy and reliability of that modules, and functionality test observation.

B. Functionality Test Results

The functionality of testing from this module have achieved more significantly. that consist of the PDF file input process is capable of displaying the page cut of the PDF, The conversion of PDF to test is functioning properly, and it is able to display the results of test conversion from PDF. The indexing process with LaMini-LM [21] or OpenAI llms provides information on the number of tokens used for embedding into Generative Pretraining Transformers (GPT) [25].

C. Performance Test Results

The performance of testing from this module for scientific articles focused on the English version, yielding specific results. One of the questions, specifically ”justification (if applicable)” showed a significantly low average from the similarity index score. During that evaluation, it was observed that the machine misinterpreted the prompt. Additionally, the engine did not directly utilize the concluding section of the source document for generating the response module.

The question types (Q) from table I that we have asked respondents were about study selection such as title, abstract, introduction, methods, results, and discussion/conclusion. Where sub items questions list that talk about relevancy to the research topic, inclusion in the review, and justification. After we collect the respondent answers, we analyze based on the average values calculated from the given table I, we can derive some analysis from the data. Here are a few observations:

1) **Variation in Average Scores**: The average scores across the articles show some variation. The average scores range from approximately 0.58 to 0.83. This indicates that there are differences in the performance or response to the questions for each article.
2) **Consistency in Art. 1**: Article 1 has the highest average score of approximately 0.83. This suggests that the responses for this article were relatively consistent and received higher scores across the questions compared to other articles.
3) **Lower Scores for Art. 5**: Article 5 has the lowest average score of approximately 0.58. This implies that the responses for this article were relatively lower or less favorable compared to the other articles.
4) **Variation in Question Responses**: Looking at the individual question scores within each article, we can identify variations in the responses. For example, in Article 1, the scores range from 0.58 to 0.83, indicating differences in the perception or evaluation of the questions within that article.
5) **Similar Scores for Questions**: Some questions across the articles have similar scores, while others show variations. Analyzing the patterns and similarities in the question scores can provide insights into the consistency or agreement among the respondents.

It’s important to note that these observations are based on the average values calculated from the given table. Further analysis and interpretation can be done based on the specific context, research objectives, and any additional information available about the study or questionnaire used.

VI. CONCLUSION

From this research, we conclude on several points. We have made the best scenario based on our latest approach which our module has better accuracy responses at five of nine questions listed based on questions about scientific article

content reviews that average similarity level agreement around 0.58 to 0.8ś. However, a limitation of that module cannot effectively perform some prompt techniques, which issue is still a development issue to solve. To overcome this problem, future research can focus more on enhancing the level of machine intelligence and refining the system interface design.

In order to improve the results, further research is necessary to include additional data from various platforms and interdisciplinary fields that intersect with the topic. It would also be beneficial to introduce variations in conditions during performance testing.

ACKNOWLEDGMENT

I would like to thank for our Institution which held on Indonesia (Universitas Hang Tuah, Universitas Negeri Yogyakarta, Politeknik Negeri Padang, dan Politeknik Elektronika Negeri Surabaya) which has provided a lot of appreciation and support for making this paper, especially our respondents who have taken much time to fill out several questionnaires as a form of our appreciation for those who have helped a lot in this study. Web Intelligence and Data Mining Lab, National Central University (NCU) which has provided many insights, sufficient ideas and resources to support our research.

REFERENCES

[1] Bramer, Wichor M., et al. "Optimal database combinations for literature searches in systematic reviews: a prospective exploratory study." Systematic reviews 6 (2017): 1-12.

[2] Moher, David, et al. "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement." Annals of internal medicine 151.4 (2009): 264-269.

[3] Turner, Rebecca M., Sheila M. Bird, and Julian PT Higgins. "The impact of study size on meta-analyses: examination of underpowered studies in Cochrane reviews." PloS one 8.3 (2013): e59202.

[4] Pussegoda, Kusala, et al. "Systematic review adherence to methodological or reporting quality." Systematic reviews 6.1 (2017): 1-14.

[5] Marshall, Iain J., Joël Kuiper, and Byron C. Wallace. "RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials." Journal of the American Medical Informatics Association 23.1 (2016): 193-201.

[6] O'Mara-Eves, Alison, et al. "Using text mining for study identification in systematic reviews: a systematic review of current approaches." Systematic reviews 4.1 (2015): 1-22.

[7] Thomas, James, John McNaught, and Sophia Ananiadou. "Applications of text mining within systematic reviews." Research synthesis methods 2.1 (2011): 1-14.

[8] Wallace, Byron C., et al. "Semi-automated screening of biomedical citations for systematic reviews." BMC bioinformatics 11.1 (2010): 1-11.

[9] Medlar, Alan, and Dorota Głowacka. "Game over? A review of gamification in information retrieval." ACM SIGIR Forum. Vol. 55. No. 2. New York, NY, USA: ACM, 2022.

[10] Alquliti, Wajdi Homaid, and Norjihan Binti Abdul Ghani. "Convolutional neural network based for automatic text summarization." International Journal of Advanced Computer Science and Applications 10.4 (2019).

[11] Lechtenberg, Fabian, et al. "Information retrieval from scientific abstract and citation databases: A query-by-documents approach based on Monte-Carlo sampling." Expert Systems with Applications 199 (2022): 116967.

[12] Bigdeli, Amin, et al. "Gender Fairness in Information Retrieval Systems." Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022.

[13] Jia, Menglin, et al. "When in Doubt: Improving Classification Performance with Alternating Normalization." Findings of the Association for Computational Linguistics: EMNLP 2021. 2021.

[14] Scells, Harrisen, et al. "The Impact of Query Refinement on Systematic Review Literature Search: A Query Log Analysis." Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval. 2022.

[15] Kusa, Wojciech, Allan Hanbury, and Petr Knoth. "Automation of Citation Screening for Systematic Literature Reviews using Neural Networks: A Replicability Study." Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I. Cham: Springer International Publishing, 2022.

[16] Fan, Yixing, et al. "Pre-training methods in information retrieval." Foundations and Trends® in Information Retrieval 16.3 (2022): 178-317.

[17] Abdullah, Mohd Hafizul Afifi, et al. "Systematic Literature Review of Information Extraction from Textual Data: Recent Methods, Applications, Trends, and Challenges." IEEE Access (2023).

[18] Wohlin, Claes, et al. "Guidelines for the search strategy to update systematic literature reviews in software engineering." Information and software technology 127 (2020): 106366.

[19] Snyder, Hannah. "Literature review as a research methodology: An overview and guidelines." Journal of business research 104 (2019): 333-339.

[20] Patil, Siba Prasad, and Rasmita Rautray. "SMATS: Single and Multi Automatic Text Summarization." Karbala International Journal of Modern Science 9.1: 6.

[21] Wu, Minghao, et al. "Lamini-lm: A diverse herd of distilled models from large-scale instructions." arXiv preprint arXiv:2304.14402 (2023).

[22] Su, Hongjin, et al. "One Embedder, Any Task: Instruction-Finetuned Text Embeddings." arXiv preprint arXiv:2212.09741 (2022).

[23] Danopoulos, Dimitrios, Christoforos Kachris, and Dimitrios Soudris. "Approximate similarity search with faiss framework using fpgas on the cloud." Embedded Computer Systems: Architectures, Modeling, and Simulation: 19th International Conference, SAMOS 2019, Samos, Greece, July 7–11, 2019, Proceedings 19. Springer International Publishing, 2019.

[24] Asyrofi, Rakha, Taufik Hidayat, and Siti Rochimah. "Comparative Studies of Several Methods for Building Simple Traceability and Identifying The Quality Aspects of Requirements in SRS Documents." 2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS). IEEE, 2020.

[25] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).