Advancing Software Engineering Research through AI-Assisted Mapping Studies

Lennart Franck (lennart.franck@hs-flensburg.de)[1]

[1] *Flensburg University of Applied Sciences (FUAS), Kanzleistraße 91-93, 24943 Flensburg, Germany*

Abstract

Context: The exponential growth in academic publications, driven by the "publish or perish" culture and recent global events like COVID-19, challenges researchers in managing large volumes of literature. Systematic mapping studies aim to provide comprehensive overviews, but are traditionally labor-intensive and prone to errors.
Objective: This thesis explores the integration of AI, through the development of the SMS Helper tool, to streamline and enhance the SMS process by automating key tasks and evaluating its performance.
Method: The SMS Helper was developed to automate research question formulation, keyword selection, paper sorting, and information extraction. Its performance was evaluated through case studies in software engineering, comparing AI-assisted results with traditional methods.
Results: The SMS Helper significantly reduced manual effort and improved consistency and accuracy. However, it highlighted the need for human oversight in complex tasks. The tool performed well in automating key SMS tasks, though challenges in keyword strategy and criteria application were noted.
Conclusion: AI integration in systematic mapping studies offers substantial efficiency and consistency benefits. While not a replacement for human expertise, AI tools like the SMS Helper can complement researchers, making systematic mapping studies more effective and scalable. Future work should focus on refining AI capabilities and enhancing human-AI collaboration.

1 Introduction

The rapid advancement of technology and the "publish or perish" culture in academia have led to an exponential increase in research publications. This trend has been further accelerated by recent global events, such as the COVID-19 pandemic, which caused a significant spike in academic submissions Müller et al. [2022]. For instance, publishers like Elsevier reported a 58% increase in new submissions from February to May 2020 compared to the same period the previous year. This surge in publications has exacerbated the challenges faced by researchers, who struggle to keep up with the growing volume of literature. As Müller et al. highlights, the influx of research, driven by pressures for career advancement and institutional funding, often leads to an increase in lower-quality studies. This scenario complicates the task of sifting through the vast literature to identify valuable content, underscoring the urgent need for more efficient literature review processes.

Systematic mapping studies (SMS) are particularly affected by these challenges, as they aim to provide comprehensive quantitative overviews of research areas. The manual effort required to conduct SMS, including repetitive tasks such as searching and sorting relevant papers, and extracting data, is becoming increasingly untenable. The mounting difficulties

in managing large datasets and ensuring the quality of reviews highlights the potential role of artificial intelligence (AI) in this domain.

To address these challenges, this thesis focuses on the development of the SMS Helper, an AI-assisted tool designed to streamline and enhance the SMS process. The tool aims to reduce manual workload, improve consistency, and increase the accuracy of systematic mapping studies. This semi-automated approach is particularly valuable in handling the vast amount of data generated by the surge in publications, thereby enabling researchers to focus on more complex analytical tasks and decision-making processes.

The central research questions (RQ) guiding this thesis are:

- RQ1: How to adjust the mapping study process when using AI?
- RQ2: How to (semi-)automate the different steps of systematic mapping studies involving generative AI?
- RQ3: How did the tool and adjusted mapping process perform in the evaluation case?

In this thesis, we explore the integration of the SMS Helper into the systematic mapping study workflow. The thesis begins with a background exploration of key AI technologies, including generative AI models like Generative Adversarial Networks and large language models such as GPT, which are integral to the tool's functionality. It also discusses the distinctions between systematic mapping studies, systematic reviews, and case studies, providing a foundation for understanding the specific contributions and challenges associated with each method.

The related work section reviews existing research on the automation of literature reviews, particularly focusing on how AI can aid in creating search strings and evaluating study relevance. This review sets the stage for understanding the current capabilities and limitations of AI in the domain of systematic reviews and mapping studies.

In examining the adjusted process, the thesis details how the SMS Helper modifies traditional SMS methodologies, addressing RQ1 on the adjustment of these processes through AI integration. The tool's capabilities in automating research question generation and keyword selection are highlighted, showcasing its potential to streamline the mapping process.

The technical description of the SMS Helper provides a comprehensive overview of its architecture and functionalities, using the 4+1 View Model. This includes discussions on the logical and development perspectives, interaction processes, and the use of advanced prompt engineering techniques. The evaluation of the tool's performance against key metrics such as accuracy and repeatability ties into RQ2, concerning the semi-automation of SMS tasks.

Through a series of case studies, the practical application of the SMS Helper is examined by re-executing selected systematic mapping studies in various software engineering domains. This empirical evaluation addresses RQ3, providing insights into the tool's strengths and areas for improvement, particularly in aligning with traditional methods in research question formulation, keyword selection, paper sorting, and data extraction.

The discussion synthesizes these findings, emphasizing the tool's role in reducing the manual effort required in systematic mapping studies and its potential to enhance consistency and accuracy. However, it also acknowledges the tool's limitations, particularly in handling complex criteria and the necessity for human oversight. The discussion suggests directions for future research and tool development, focusing on refining AI capabilities and better integrating them with human expertise.

2 Background

2.1 Generative AI

Generative AI represents a significant advancement in AI, focused on creating models that can generate new content, such as text, images, or even complex simulations. Among the most notable types of generative AI are Generative Adversarial Networks (GANs), language models (such as GPT), and various forms of generative models used in natural language processing.

GANs, introduced by Goodfellow et al. [2014], operate through a dual-model system comprising a generator and a discriminator. The generator creates synthetic data, while the discriminator evaluates its authenticity. The iterative training process, where the generator aims to produce data indistinguishable from real data and the discriminator aims to improve its detection, results in high-quality synthetic data generation.

Large Language models like GPT (Generative Pre-trained Transformer), developed by OpenAI, are another prominent form of generative AI. These models leverage large-scale unsupervised learning to pre-train on a diverse corpus of text, followed by fine-tuning on specific tasks [Radford et al., 2018]. GPT-4, for instance, is capable of generating human-like text, performing translation, answering questions, and even writing code. Its architecture is based on the transformer model, which uses self-attention mechanisms to manage dependencies across long sequences, making it highly effective for various natural language processing tasks.

Generative AI also intersects with Retrieval-Augmented Generation (RAG), which combines the strengths of retrieval-based and generative models [Gao et al., 2024]. RAG architectures, such as those used in systems like LangChain [LangChain], leverage external knowledge bases during the generation process, allowing for more accurate and contextually relevant outputs. This hybrid approach addresses the limitations of purely generative models, particularly in domains requiring up-to-date or highly specific information.

In summary, generative AI encompasses various advanced models and techniques, from GANs and GPT to RAG systems, each contributing uniquely to the field of AI. These technologies not only push the boundaries of what AI can generate, but also significantly impact practical applications in software engineering and beyond.

2.2 Understanding systematic maps

Systematic mapping studies (SMS), systematic reviews (SR), and case studies are distinct research methodologies, each serving different purposes and employing unique processes. SMS aim to provide an overview of a research area, identifying the quantity, type, and distribution of research conducted within it. The primary goal is to classify existing literature into a structured map, often represented visually, which highlights trends and gaps in the research field Petersen et al. [2008].

In contrast, SR delve deeper into the literature, focusing on synthesizing findings from multiple studies to answer specific research questions. SRs employ rigorous methodologies to minimize bias, including comprehensive search strategies, inclusion and exclusion criteria, and detailed analysis of the included studies. This approach allows for more robust conclusions and meta-analyses, providing high levels of evidence on a particular topic Kitchenham et al. [2010].

Case studies, on the other hand, are empirical investigations that explore contemporary phenomena within real-life contexts. They are typically qualitative, offering in-depth insights into specific instances rather than generalizable findings. Case studies are valuable

for understanding complex issues in detail, generating hypotheses, and providing rich contextual analysis Flyvbjerg [2011].

The differences between these methodologies are evident in their goals and applications. SMS are broader and less detailed, focusing on categorization and high-level trends, making them suitable for new or fragmented research areas where a comprehensive overview is needed. SRs are more detailed and quantitative, ideal for mature fields where synthesizing specific findings can influence practice and policy. Case studies are best suited for exploring unique or complex cases in depth, often informing theory development and practical implementation.

The systematic mapping process proposed by Petersen et al. is widely recognized as the default methodology for conducting SMSs in software engineering. This process is designed to provide a comprehensive overview of a research area, categorizing existing literature into a structured framework that highlights trends, gaps, and research activity distribution. This structured approach not only facilitates a high-level understanding of the research area but also identifies under-explored topics and emerging trends, making it an essential tool for researchers aiming to summarize and analyze large volumes of academic literature.

3 Related Work

AI applications in literature reviews have rapidly evolved to address the challenges posed by the sheer volume of academic publications. The prevalent "Publish or Perish" culture in academia, where a high volume of publications is often key to career advancement, has inadvertently contributed to an influx of lower-quality research. This situation creates a fertile ground for AI applications that promise to enhance the speed, accuracy, and cost-efficiency of literature reviews, as noted by Müller et al. [2022] who highlights the mounting challenges faced by the research community. In addition, it is important to note that the application of AI in literature studies is a relatively new field, with limited studies published to date.

Search

Research on the formulation of search strings, particularly Boolean search strings, has garnered significant attention in the context of automating SRs. One notable work in this domain is by Alshami et al. [2023] who in their study emphasize the potential of ChatGPT to automate various steps in the SR process, including the creation of Boolean queries. They propose that ChatGPT can be used to generate and refine search strings, potentially reducing human error and effort in the systematic review process.

Additionally, Claveau [2022] delves in his paper into the use of neural networks for expanding queries in information retrieval tasks. He highlights how neural text generation can improve the comprehensiveness of search strings, thereby enhancing the retrieval of relevant literature. Claveau's research demonstrates that leveraging neural networks for query expansion can significantly improve search results, which is crucial for the accuracy and completeness of systematic reviews.

Moreover, Li et al. [2020] provide a comprehensive overview of tools designed to automate the entire query formulation process for SR in their paper. They examine various automation tools that facilitate the generation of Boolean queries, highlighting their strengths and limitations. Li et al. discuss how these tools integrate with different stages of the SR process, from initial query formulation to final literature selection, thereby streamlining the workflow and reducing manual workload.

In another study Wang et al. [2023] specifically investigate the capability of ChatGPT to generate effective Boolean queries for SR. Their paper "Can ChatGPT write a good

Boolean query for systematic review literature search?" assesses the quality of queries generated by ChatGPT compared to those crafted by domain experts. They find that while ChatGPT shows promise in formulating Boolean queries, there are areas where human oversight is still necessary to ensure precision and relevance.

Noroozi et al. [2023] describe an AI-based methodology for automating SR, which refines search queries and classifies documents to improve the efficiency and quality of reviews. This approach uses NLP-based models to iteratively refine search terms and ML classifiers to streamline the initial document evaluation, allowing researchers to focus more on the substantive aspects of their studies. This methodology is significant because it addresses both the scalability and precision challenges inherent in traditional SRs. By employing iterative refinement and classification, the system reduces the human workload and minimizes the risk of overlooking critical literature, which is often a limitation in manual reviews.

Relevance

Several case studies illustrate the practical applications of AI in this context. For example, Gurcan and Sevik [2019] utilized advanced AI techniques such as Latent Dirichlet Allocation (LDA) to delve into decades of Human-Computer Interaction literature, uncovering significant research trends through probabilistic topic modeling. This approach demonstrated how AI could synthesize large volumes of data into coherent themes and trends, providing a high-level overview of the research landscape. Jenko et al. [2024] tested AI platforms like the-literature.com and OpenAI's GPT-4 to synthesize literature reviews within the medical domain, evaluating their performance on accuracy and comprehensiveness, which highlighted mixed results and emphasized the need for human oversight in AI-generated summaries. The study underscores the current limitations of AI, particularly in terms of nuanced understanding and contextual interpretation, which are areas where human expertise remains indispensable. Similarly, Kempeneer et al. [2023] made use of the AI-driven tool ASReview to expedite the screening process in a SR of legal literature on open government data, showcasing significant time savings over traditional manual methods. This case study highlights the efficiency gains AI can bring to the initial stages of the review process, allowing researchers to allocate more time to in-depth analysis and interpretation.

Despite these advancements, none of the reviewed papers have implemented a semi-automated process for SMS. SMS, which aim to provide a broad overview of research trends and gaps within a specific field, can greatly benefit from AI-enhanced methodologies. However, the current literature primarily focuses on SRs or manually intensive processes. This gap presents an opportunity for future research to explore the integration of semi-automated approaches, leveraging AI to assist but not completely replace human judgment and expertise in mapping studies. By doing so, the process can combine the scalability and efficiency of AI with the critical thinking and contextual understanding of human reviewers, potentially leading to more comprehensive and accurate SMS.

4 Part I - Adjusted Process

The workflow for SMS is significantly enhanced and streamlined with the SMS Helper, which automates and simplifies key steps, reducing the manual effort required. Below, we detail the adjusted workflow as seen in figure 1 based on Petersen et al. [2008]. The focus relies on how the tool assists in constructing of a search string, sorting papers and extraction. This section will highlight the differences between the process proposed by Petersen et al. [2008] and the new, semi-automated approach enabled by the SMS Helper.

In the traditional systematic mapping study process, as outlined by Petersen et al. [2008],

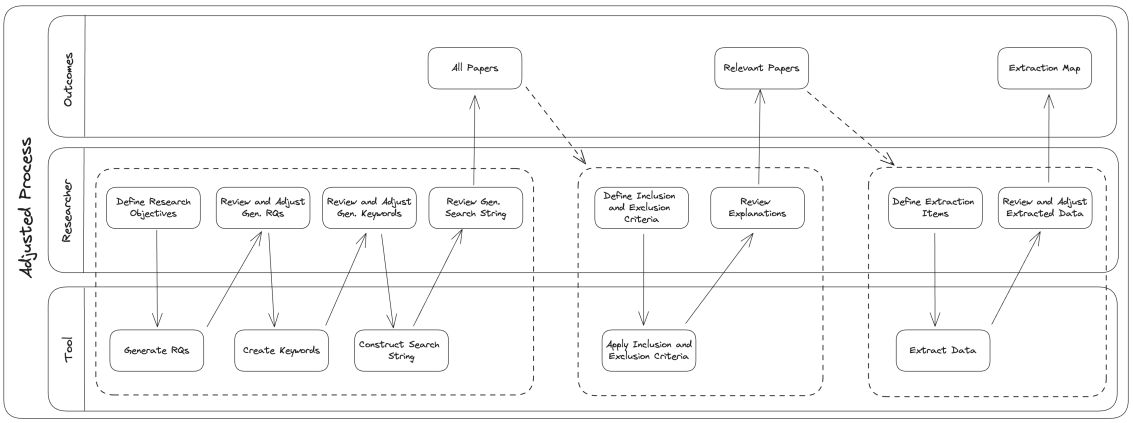


Figure 1: Adjusted process based on Petersen et al. [2008]

researchers begin by manually defining research questions based on their understanding of the research domain and objectives. This involves considerable effort in brainstorming, reviewing existing literature, and iteratively refining the questions to ensure they cover the necessary scope. This initial stage can be time-consuming and requires significant domain expertise. In contrast, the SMS Helper automates the generation of research questions from the research objective. The tool assists users by suggesting preliminary research questions based on the input objectives. Leveraging natural language processing and pre-defined templates, the tool significantly reduces the time and effort required in this stage. Researchers can then edit these questions or add their own, ensuring that the generated questions align closely with their research goals.

Once the research questions are defined, the traditional approach involves manually creating a list of keywords derived from these questions. Researchers must have a profound understanding of the domain to ensure that all relevant terms are included, and this process is often iterative, requiring multiple refinements through trial and error. This manual process can be labor-intensive and prone to oversight. With the SMS Helper, the creation of keywords is automated. The tool uses the research questions and objectives to suggest a comprehensive list of potential keywords. This automation ensures that the keyword creation process is thorough and efficient, reducing the likelihood of missing critical terms.

Constructing the boolean search string traditionally involves manually combining keywords using boolean operators (AND, OR, NOT) to create a search string that is both comprehensive and focused. Researchers often use frameworks like PICO (Problem, Intervention, Comparison, Outcome) proposed by Kitchenham and Charters [2007] to structure their search strings, but this process is manual and time-consuming. The SMS Helper enhances this step by categorizing selected keywords into the PICO framework and automatically constructing the boolean search string. This automated construction ensures that the search string is precise and comprehensive, covering all necessary aspects of the research questions.

In the traditional SMS process, sorting papers involves manually applying inclusion and exclusion criteria to each identified study. This step requires researchers to carefully review the titles, abstracts, and sometimes full texts of papers to determine their relevance to the research questions. This manual sorting process is labor-intensive and time-consuming, often leading to inconsistencies due to varying interpretations of the criteria. The SMS Helper automates this sorting process by using predefined criteria set by the researchers. The tool systematically applies these criteria to the metadata of the papers, such as titles, abstracts, and keywords, providing decisions along with confidence scores and explanations for each

inclusion or exclusion. This automation ensures a consistent and unbiased application of criteria, significantly reducing the time and effort required from researchers. However, it also highlights the importance of well-defined criteria, as the tool's decisions are based solely on the available text data.

Similarly, the extraction phase traditionally requires researchers to manually sift through selected papers to extract relevant data points, such as study outcomes, methodologies, or other specific information required for the mapping study. This process can be particularly tedious and prone to human error, as it involves interpreting and summarizing complex information. The SMS Helper facilitates this step by using RAG to automatically extract relevant data based on predefined extraction items. Researchers can specify the types of data they are interested in, and the tool will extract and compile this information into a structured format, such as a CSV file. This automated extraction accelerates the process and enhances the accuracy and consistency of the data collected, as it minimizes the subjective interpretation involved in manual extraction. However, it is essential for researchers to review and verify the extracted data to ensure its relevance and accuracy, particularly for nuanced or complex extraction tasks.

To maximize the effectiveness of the SMS Helper, it is essential to follow certain guidelines. For defining clear and concise research objectives, best practices include ensuring that the objectives are specific, relevant, and feasible. Specificity involves avoiding vague or overly broad objectives, while relevance ensures alignment with current needs and gaps in the research field. Feasibility considers the scope and practicalities within the available resources and time frame. In addition to automation, validation and iteration are crucial. The SMS Helper supports these by allowing researchers to review and refine the generated outputs at each step, ensuring accuracy and relevance. This iterative feedback loop is essential for maintaining high standards in SMS.

Despite the significant advantages, challenges and limitations exist. The SMS Helper relies on the quality of input data and predefined templates, which might not capture all the nuances of every research field. Furthermore, while the tool reduces manual effort, it cannot entirely replace the critical thinking and expertise of researchers. User feedback and continuous improvement are vital. By collecting feedback from researchers using the tool, developers can refine and enhance the SMS Helper, addressing any limitations and expanding its capabilities to better serve the research community.

By integrating the SMS Helper into the SMS workflow, researchers can significantly enhance the efficiency and effectiveness of their studies. The automation provided by the tool reduces the time and effort required and ensures more consistent and reliable results compared to the traditional process proposed by Petersen et al.. This adjusted workflow represents a significant advancement in the methodology of systematic mapping studies, leveraging AI to address the limitations of manual processes.

4.1 Answering RQ1

The integration of AI, specifically through the SMS Helper, significantly transforms the traditional systematic mapping study process. The key adjustments required when using AI focus on shifting from a manual, labor-intensive workflow to a more strategic and supportive role for researchers.

In the traditional process, researchers manually define research questions, constructing Boolean search strings, and filter relevant studies. Each of these steps demands extensive domain expertise, significant time investment, and meticulous attention to detail. This manual approach often results in variable quality due to human error and oversight.

With the adoption of AI, the workflow is streamlined, reducing the manual effort

required at each stage. The SMS Helper automates the generation of research questions from the research objective, the selection of keywords, and the construction of Boolean search strings. However, this automation necessitates a paradigm shift for researchers. They must now focus on providing high-quality inputs and critically evaluating the AI-generated outputs.

The new process emphasizes the importance of well-defined and precise research objectives. Researchers must articulate their objectives clearly and comprehensively, as the AI relies on these inputs to generate relevant research questions. The quality of the generated questions directly correlates with the specificity and clarity of the initial objective. This step requires researchers to invest more thought and effort into formulating their objectives, ensuring they are specific, relevant, and feasible.

When applying criteria to filter paper information, the role of the researcher shifts from manually reviewing each paper to defining comprehensive and detailed criteria. The SMS Helper uses these criteria to classify the papers, and the quality of the results depends heavily on how well the criteria are specified. Researchers must ensure that the criteria are exhaustive and precise, covering all necessary aspects to achieve accurate classification. This shift requires researchers to think more strategically about the inputs they provide, ensuring they encompass all relevant dimensions of the studies being evaluated.

The new process also involves more critical evaluation of AI outputs. Researchers must review the AI-generated outputs and explanations to ensure they align with the research goals. This iterative feedback loop between the researcher and the AI tool ensures that the outputs meet the required standards of quality and relevance. Another critical aspect of the adjusted process is the focus on traceability and transparency. The SMS Helper provides explanations and text references for each automated decision, allowing researchers to understand and verify the reasoning behind the results. This transparency is crucial for maintaining trust in the system and ensuring the credibility of the research findings.

Overall, the adjusted mapping study process when using AI requires researchers to adopt a more supportive and supervisory role. They must ensure high-quality inputs, critically evaluate AI outputs, and focus on strategic aspects of the research workflow. This shift reduces the manual burden on researchers while enhancing the efficiency, consistency, and reliability of systematic mapping studies. The new process leverages the strengths of AI to automate routine tasks, allowing researchers to concentrate on higher-level analytical and evaluative activities, ultimately leading to more robust and credible research outcomes.

5 Part II - Tool

5.1 Description of the Technical Solution

The SMS Helper aims to streamline the systematic mapping study process for software engineering researchers. By semi-automating the creation of research questions, applying criteria to sort, and extracting information from studies, it addresses the time-consuming nature of traditional methods. The SMS Helper will be described using the 4+1 View Model, providing a comprehensive perspective on its architecture and functionality.

Logical View

The Logical View of the SMS Helper outlines the core functionalities that collectively automate key stages of the SMS process, enhancing efficiency and accuracy for software engineering researchers.

The first functionality, Research Question and Search String Creation, assists users in defining research objectives, formulating research questions, and constructing Boolean

search strings based on selected keywords as seen in figure 2. This process leverages an interactive interface that guides users through the stages of question formulation and keyword selection, ensuring a comprehensive and accurate search strategy.

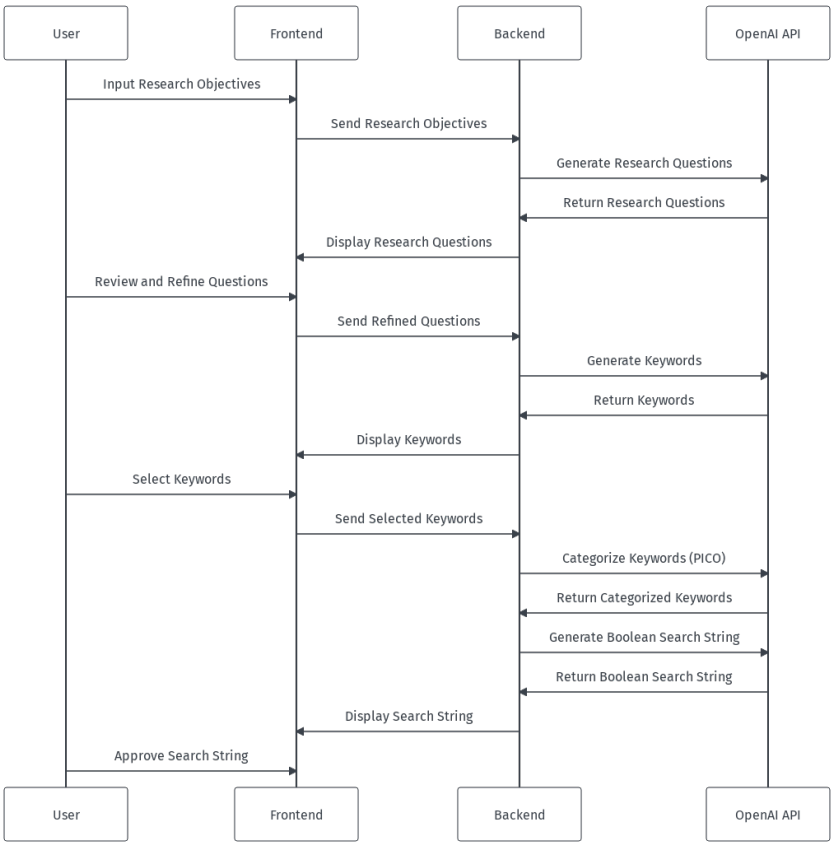


Figure 2: Search String Sequence Diagram

The second functionality involves applying Criteria on Paper Information. Users can upload a CSV file containing Paper details, which the tool then uses to apply user-defined criteria. The SMS Helper provides decisions, explanations, and confidence scores for each criterion for each paper. This streamlines the selection process afterward. This mechanism is visually depicted in the figure 3.

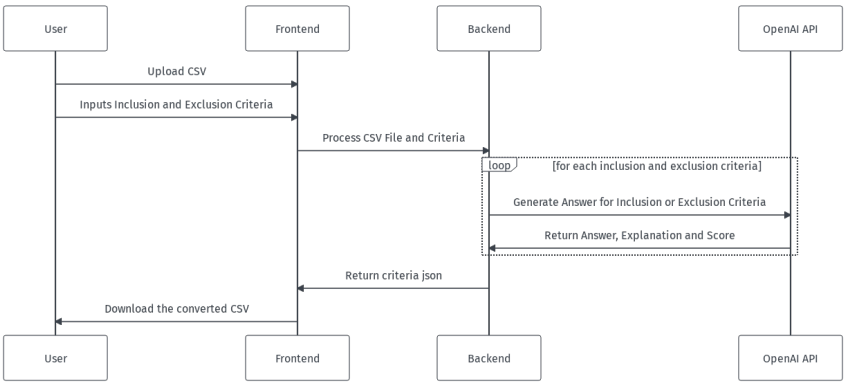


Figure 3: Applying Criteria Sequence Diagram

The final functionality, Extracting Information based on Extraction Items (Figure 4),

enables users to specify Directories containing PDFs and extraction items. The tool extracts the relevant data and generates a CSV file with the results. This process is essential for extracting information from relevant PDFs and processing them into a structured form, so the user can create a systematic map based on the results.

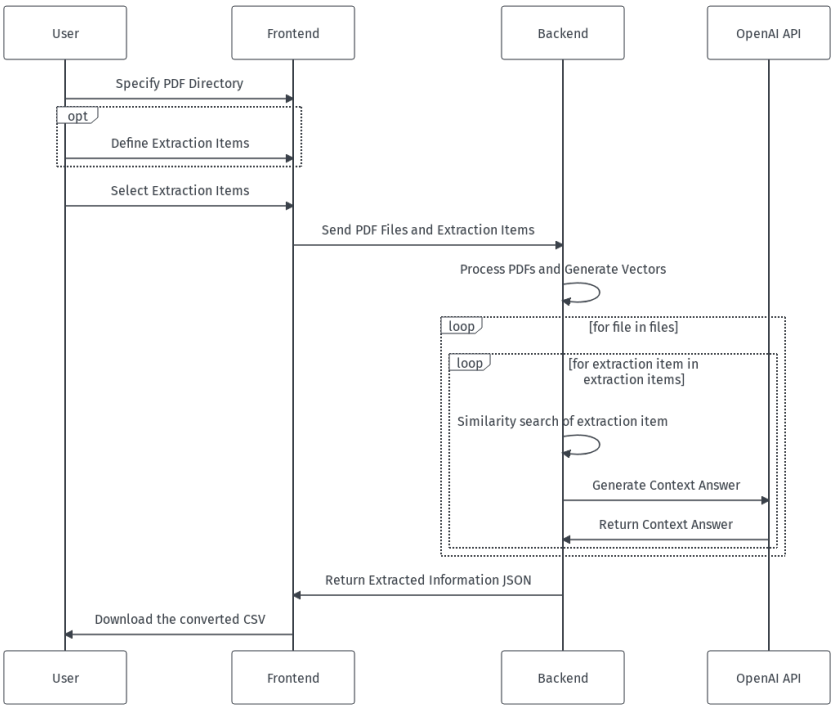


Figure 4: Extraction Sequence Diagram

Development View

The Development View of the SMS Helper offers a detailed perspective on its architectural components and their interactions. Implemented primarily in TypeScript, the system is divided into frontend and backend components, with external services integrated through API calls, as depicted in the provided component diagram seen in figure 5.

The frontend of the SMS Helper is built using Vue.js and Nuxt, ensuring a responsive and interactive user interface. At the core of the frontend is the Nuxt Component, which manages routing and rendering of pages. It utilizes the PapaParse Library for parsing CSV files, enabling users to upload and manage study information efficiently. Additionally, the NuxtUI Library provides a suite of UI components that ensure a consistent and user-friendly interface.

On the backend, also developed in TypeScript, the Server Component serves as the central piece, handling incoming HTTP requests from the frontend and orchestrating various backend tasks. The backend employs the LangChain Library to manage the creation and retrieval of vectors and interactions with OpenAI's GPT-4 API, facilitating tasks such as research question generation and data extraction. Additionally, the Formidable Library is used to handle file uploads, supporting the backend's ability to process various uploaded PDFs.

To leverage advanced AI functionalities, the system integrates external services, specifically the OpenAI API Component. This external service, accessed through API calls made by the Server Component, powers the generative AI capabilities of the SMS Helper, enabling sophisticated text processing and data extraction. The interaction between these

components is designed to be seamless and efficient. HTTP requests act as the communication bridge between the frontend and backend, while API calls connect the backend with external OpenAI services. This modular design ensures that each component can perform its specific role effectively, contributing to the overall functionality and performance of the SMS Helper.

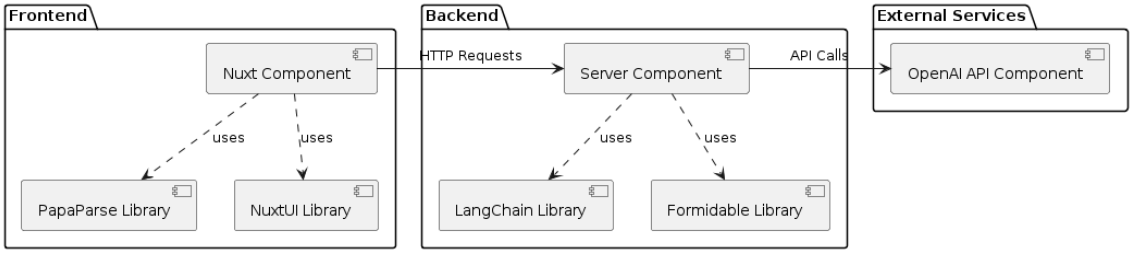


Figure 5: Component Diagram

Process View

The Process View of the SMS Helper illustrates how various components interact during different stages of the SMS process. This view is crucial for understanding the dynamic behavior of the system as users perform tasks such as defining research questions, sorting papers, and extracting information from documents. The following sections describe these processes in detail, supported by the provided activity diagrams.

The search string process begins when the user inputs the research objective into the system. The SMS Helper then generates a set of research questions based on this input. If the user is not satisfied with the generated questions, they can adjust them until they meet their requirements. Once the research questions are finalized, the system generates relevant keywords. The user selects the most appropriate keywords, and the system constructs a Boolean search string for literature searches. This sequence of interactions is depicted in figure 6, which shows the flow from inputting objectives to generating the final search string.

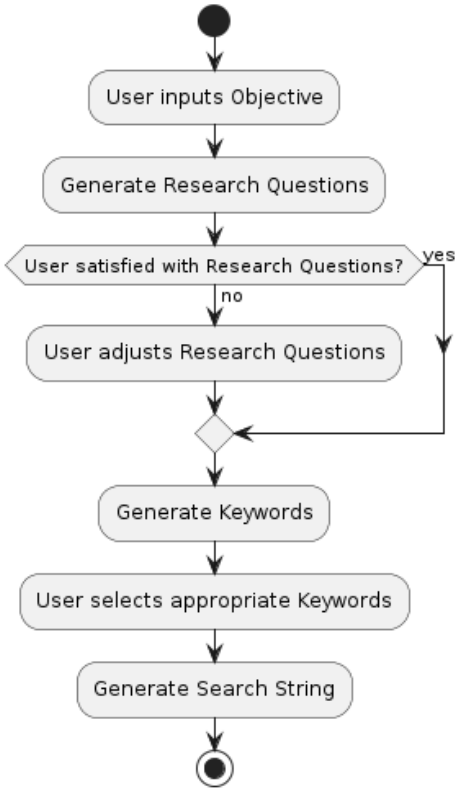


Figure 6: Search String Activity Diagram

For applying criteria, the user uploads a CSV file containing study details. The system first checks if the file format is correct. If the file is correctly formatted, the user defines inclusion and exclusion criteria for the studies. The SMS Helper applies these criteria to filter the studies and provides decisions, explanations, and confidence scores. Finally, the user can download a CSV file containing the filtered studies. This workflow is detailed in figure 7, highlighting the decision points and actions taken during the filtering process.

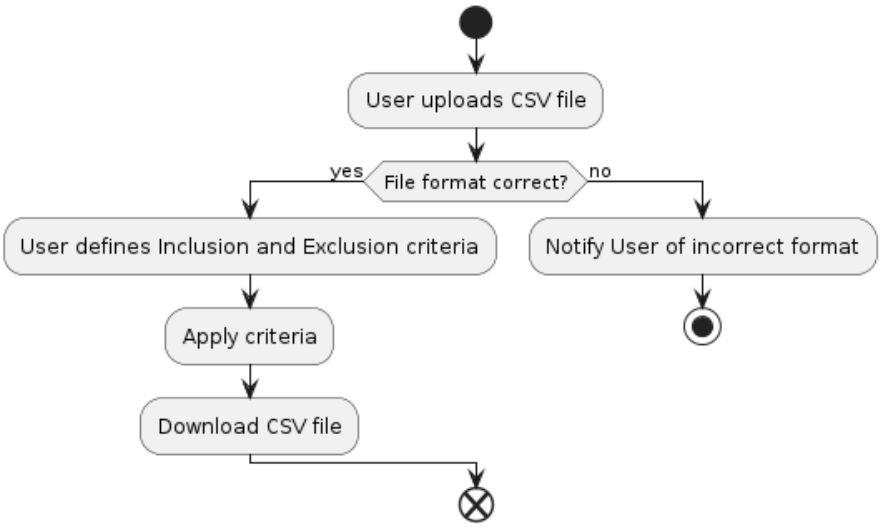


Figure 7: Criteria Activity Diagram

The final process shown in figure 8 involves extracting specific information from selected PDFs. The user uploads a folder containing the relevant papers, and the system verifies that the folder contains the correct file formats. If the files are valid, the user can choose to create new extraction items or select existing ones from a table. The SMS Helper then processes the files according to the selected extraction items and generates a CSV file with the extracted data. The user can download this file once the process is complete. If the folder contains incorrect file formats, the system notifies the user.

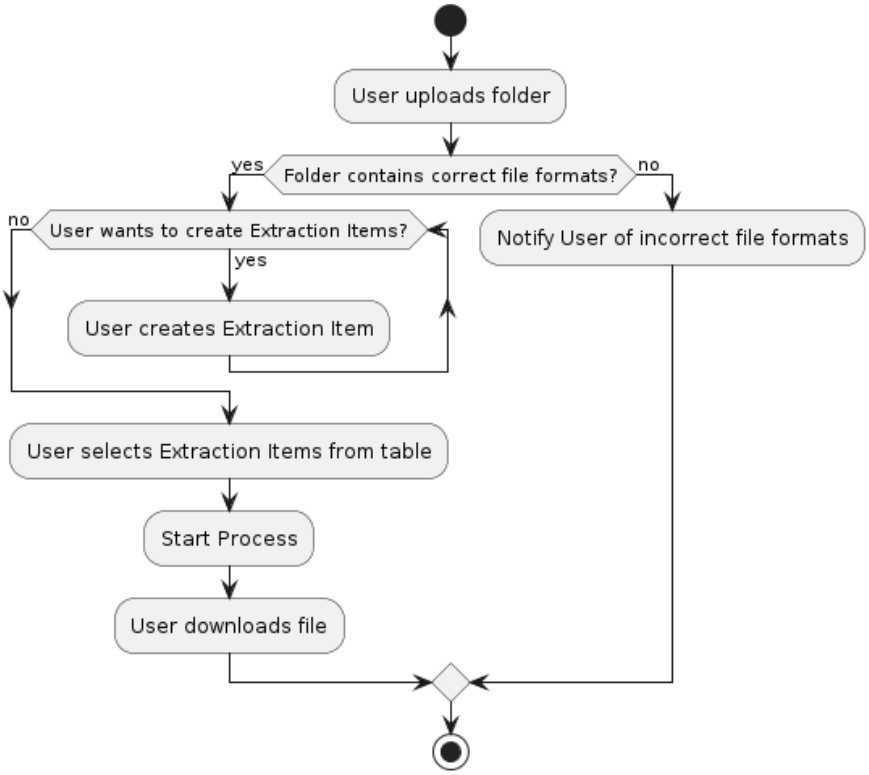


Figure 8: Extraction Activity Diagram

Physical View

The Physical View of the SMS Helper provides an overview of the system's deployment architecture, illustrating how the various components are distributed and interact within the physical infrastructure.

In this architecture, as seen in figure 9, the client interacts with the SMS Helper Server through the Web. The server is divided into two primary components: the Nuxt Frontend Component and the Nuxt Backend Component. The Nuxt Frontend Component handles the user interface, serving the necessary frontend code, including HTML, CSS, and JavaScript, to the client. This component ensures that users have a responsive and interactive experience while interacting with the SMS Helper. The Nuxt Backend Component processes the application logic and handles data management. It communicates with the frontend component through HTTP calls, processing user inputs, managing data flows, and performing essential computations to support the system's functionality.

Additionally, the backend component interfaces with an external service, the GPT-4o API hosted on the OpenAI server. The backend makes HTTP calls to the OpenAI server to leverage its advanced AI capabilities, such as generating research questions and extracting relevant information from documents.

A Nuxt application can be hosted in various ways to suit different deployment needs, including traditional server hosting, static site generation, and modern serverless or edge environments.

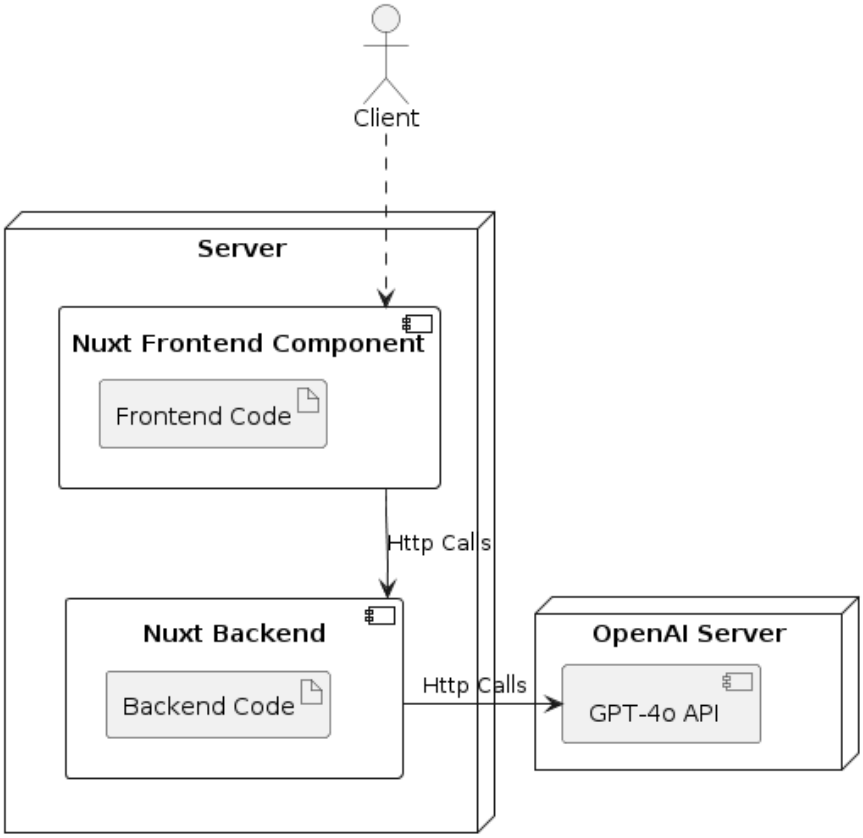


Figure 9: Deployment Diagram

Scenarios

The Scenario View of the SMS Helper provides an illustrative perspective on how researchers interact with the system through various use cases. The use case diagram seen in figure 10 offers a clear visualization of the primary functions that the software supports and the user's role in these processes.

In this Scenario View, the main actor is the researcher, who utilizes the SMS Helper to perform key tasks essential to conducting systematic mapping studies. The use case diagram outlines three primary use cases:

1. Creating a Search String: This use case involves the researcher defining the research objective and generating a corresponding search string. The process begins with the researcher inputting the research objectives, followed by the system generating relevant research questions and keywords. The researcher then refines these keywords to form an effective Boolean search string.

2. Applying Criteria: In this use case, the researcher uploads a CSV file containing study details. The system verifies the file format and allows the researcher to define inclusion and exclusion criteria. The SMS Helper then applies these criteria to filter the studies, providing decisions, explanations, and confidence scores. The researcher can download a filtered CSV file with the relevant studies.

3. Extracting Information: This use case involves the researcher uploading a folder with documents from which information needs to be extracted. The system checks the file formats and enables the researcher to create or select extraction items. Once the process is initiated, the SMS Helper extracts the relevant data and generates a CSV file that the researcher can download.

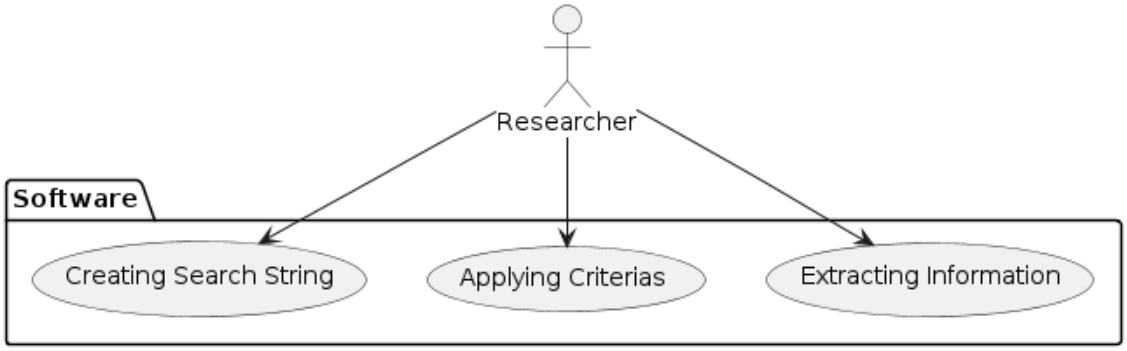


Figure 10: Use case Diagram

5.2 Prompt Engineering

Prompts in the SMS Helper are carefully crafted to ensure accurate and relevant outputs for different stages of the systematic mapping process. The prompts are designed using iterative testing in the OpenAI Playground, incorporating methodologies like prompt chaining and few-shot examples to enhance their effectiveness.

In the following section, I will provide a detailed description of each prompt used in the SMS Helper. For each prompt, I will explain its purpose and how it contributes to the SMS process. This detailed breakdown will help in understanding the role of each prompt and how it facilitates the automation and enhancement of research tasks within the tool.

Research Question Prompt

```
Given the objective "{objective}" please develop four research
    questions.

Research Questions have the following characteristics:
- In general, a good research question should be appropriate,
   meaningful, and purposeful
- Provide an overview of a research area and identify the
   quantity and type of research and results available within
   it.
- Focused on the topic area, venues, research methods or
   trends
- Be short sentenced and not to detailed

Please follow the characteristics and only provide the
   question in JSON format with the keys 'Id' and 'Question'
   and root key 'Questions'.
```

This prompt is designed to facilitate the generation of research questions based on a given objective. By leveraging the power of few-shot examples, the prompt aims to produce

high-quality research questions that align with the specified characteristics. The use of Few-Shot Prompting helps the model understand the format and nature of the desired output, thereby improving the relevance and appropriateness of the generated questions.

Keywords Prompt

```
Follow my instructions precisely to develop a highly effective
    Boolean query for a systematic mapping study literature
   search. Do not explain or elaborate. Only respond with
   exactly what I request. First, Given the following
   objective and research questions please identify 20 terms
   or phrases that are relevant. The terms you identify should
    be used to retrieve relevant studies, so be careful that
   the terms you choose are not too broad. You are not allowed
    to have duplicates in your list. Keywords are most likely
   just one or two words.
Please only give the list of keywords without hyphens or
   commas.

Objective: {objective}

Research Questions:
{research_questions}
```

This prompt instructs the LLM to generate a list of 20 precise and relevant keywords or phrases based on a given objective and set of research questions. These keywords are intended for use in a systematic mapping study literature search, ensuring that they are neither too broad nor duplicated. The LLM is guided to provide only the list of keywords in a simple format, without additional explanations or punctuation. This prompt uses Zero-Shot Prompting for generating results.

Categorizing Prompt

```
Task: Categorize the following keywords into one of the PICO
   categories: Problem (P), Intervention (I), Comparison (C),
   Outcome (O).

PICO Definitions for Software Engineering:
- Problem (P): Keywords related to the issue, challenge, or
   context being addressed. Examples: software bugs,
   performance bottlenecks, legacy code maintenance.
- Intervention (I): Keywords related to the technique, method,
    tool, or approach being studied to address the problem.
   Examples: automated testing, agile methodology, refactoring
   .
- Comparison (C): Keywords related to the alternative to the
   intervention. Examples: manual testing, waterfall model, no
    optimization.
- Outcome (O): Keywords related to the results or effects
   being measured. Examples: reduced defect rate, increased
   development speed, enhanced security.
- When an item does not fit one of these categories, mark it
   as (N/A).

Instruction: For each keyword provided, categorize it into one
    of the PICO categories by writing the appropriate letter (
   P, I, C, O, N/A) next to the keyword.
keywords:
{keywords}
```

Search String Prompt

```
Using the categorized keywords, create a Boolean query that
   can be
submitted to IEEE or Scopus which groups together keywords
   from each category.
For example:
((itemP1[Title/Abstract] OR itemP2[Title/Abstract] or itemP3[
   Title/Abstract]) AND
(itemI1[Title/Abstract] OR itemI1[Title/Abstract] OR itemI1[
   Title/Abstract]) AND
(itemC1[Title/Abstract] OR itemC2[Title/Abstract] OR itemC3[
   Title/Abstract])) AND
(itemO1[Title/Abstract] OR itemO2[Title/Abstract] OR itemO3[
   Title/Abstract])

Please only provide the search string.

categorized list: {keywords}
```

These two prompts are used together to simplify a complex task into manageable subtasks through prompt chaining. The whole idea is based on Wang et al. [2023] who used this multistep prompt to create search strings for medical databases.

The first prompt categorizes a given list of keywords into one of the PICO (Problem, Intervention, Comparison, Outcome) categories. The definitions provided for each category help ensure that the keywords are accurately classified. The LLM is instructed to label each keyword with the appropriate category letter (P, I, C, O, or N/A) based on the definitions and examples given. Few-Shot Prompting is used to give the LLM a better understanding of the PICO categories.

The second prompt takes the categorized keywords and constructs a Boolean query suitable for submission to databases like IEEE or Scopus. The query groups keywords from each PICO category together, using a specific format that ensures each keyword is searched for within the Title/Abstract fields. The prompt instructs the LLM to only provide the resulting search string, formatted to combine keywords from each category using logical OR operators within categories and logical AND operators between categories. Few-Shot Prompting is used to let the LLM know how to structure the Boolean search string.

Criteria Matching Prompt

```
You will be given a text which consists of a title, year,
   keywords and an abstract. You will also be given a criteria
   . You have to decide if the text meets the criteria or not.

Criteria: {criteria}

Text: {text}

Only answer in JSON with the following keys:
- "decision" which is only yes or no
- "explanation" which is a short explanation that includes
   text citations for why you made your decision
- "score" the probability between 0.0 and 1.0 that your answer
    is correct.
```

This prompt is designed to evaluate whether a given text meets specified criteria. The text includes a title, year, keywords, and an abstract, which beforehand is computed as one on the backend. The LLM must evaluate the text against the provided criteria and respond in JSON format with three keys: "decision" a simple yes or no answer indicating if the criteria are met, "explanation" a brief justification that includes citations from the text, and "score" a probability between 0.0 and 1.0 indicating the confidence in the decision. Tian et al. [2023] found that LLMs can express their uncertainty with numerical probabilities as well as or better than with words.

Extraction Prompt

```
You are an assistant for question-answering tasks for study
   extraction items.
Use the following pieces of retrieved context to answer the
   question and use the given return types. If you don't know
   the answer return 'not sure'. Only answer with the
   information in the return type format.
Question: {question}
Return Type: {return_type}
Context: {context}
Answer:
```

This prompt is intended for a RAG task, where the LLM assists with question-answering based on provided study extraction items. The LLM is given a specific question, the desired return type, and relevant context which is extracted using similarity search on a vector store. Using this information, the LLM must generate an answer strictly based on the retrieved context. If the LLM cannot answer the question with the found context, it should respond with 'not sure'. The response must be formatted according to the specified return type, ensuring that the information is precise and adheres to the required format. The consequence is that the LLM cant create an own response that is not within the context and therefore is false.

5.3 User Interface Workflow

To provide a comprehensive understanding of the SMS Helper tool, this section outlines the user interface (UI) workflow through a series of screenshots. These images illustrate the key steps a user takes to utilize the tool's functionalities, from search to data extraction.

The user begins at the search page of the SMS Helper tool, where they input the project objectives and are guided through the process of generating research questions (RQs) and keywords. This page is designed to streamline the initial stages of a systematic mapping study, making it easier for researchers to define their research scope and focus. Figure 11 shows the search page of the SMS Helper tool.

After generating the research questions and keywords, the user navigates to the sorting page. Here, they can upload a CSV file containing study details and define the inclusion and exclusion criteria. The tool then sorts the papers based on these criteria, providing decisions with explanations and confidence scores. This functionality significantly reduces the manual effort required for literature screening. Figure 12 illustrates the sorting page where users select a CSV file and add criteria.

In the final step, users move to the extraction page. This page allows users to specify extraction items and extract relevant data from the selected papers. The tool processes the documents and presents the extracted data in a structured format, facilitating analysis and interpretation. Figure 13 displays the extraction page where users can select a directory, add extraction items, and select them.

These figures collectively illustrate the seamless workflow of the SMS Helper tool, highlighting its user-friendly design and functionality that assist researchers in conducting SMS efficiently.

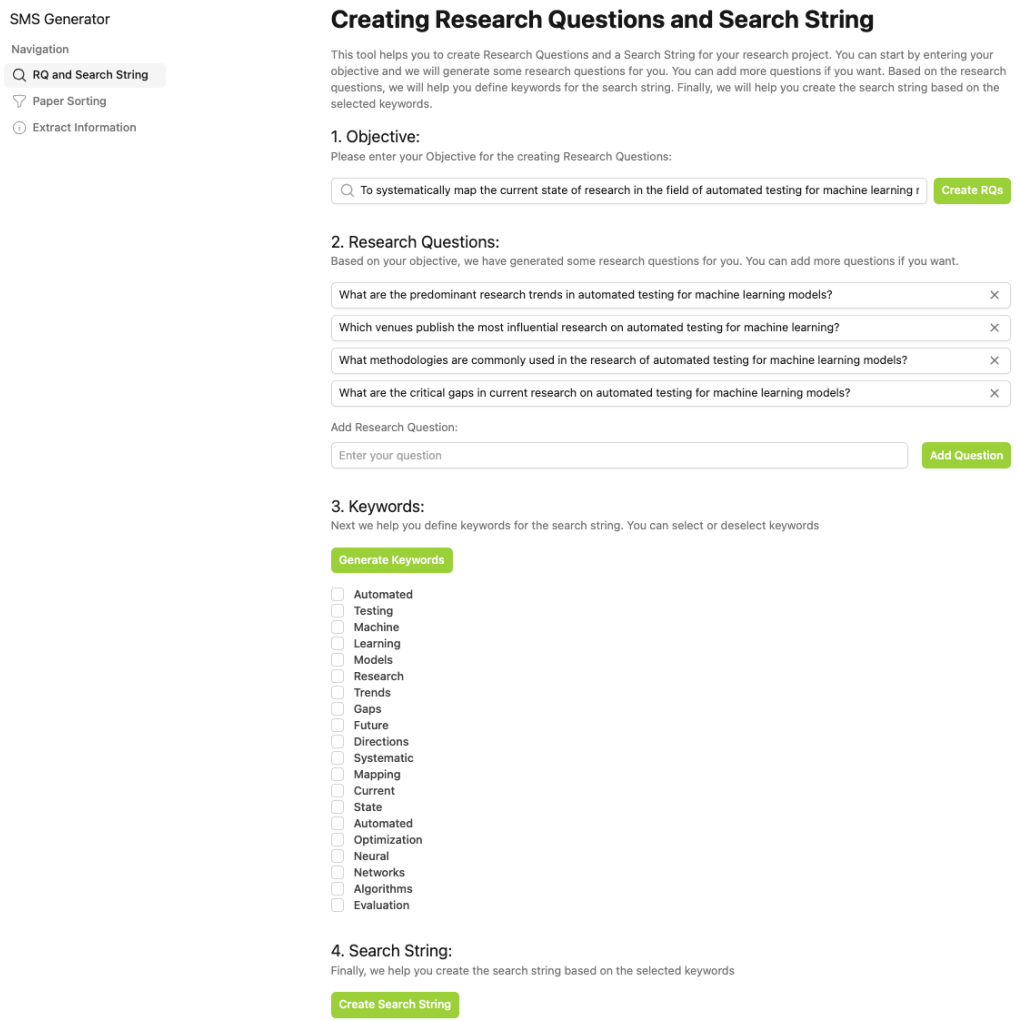


Figure 11: Search Page of the SMS Helper Tool

5.4 Key metrics

To critically evaluate the SMS Helper, it is essential to identify and measure key metrics that determine the system's effectiveness and reliability. These metrics provide a foundation for assessing the quality and performance of the tool, ensuring it meets the rigorous demands of systematic mapping studies in software engineering.

First, the accuracy of the SMS Helper's outputs is crucial. This involves assessing how well the tool generates research questions, filters relevant literature, and extracts information compared to manual processes. High accuracy indicates reliable and precise outputs, essential for credible research findings.

Repeatability is another critical metric, referring to the tool's ability to produce consistent results when the same input data and criteria are used multiple times. This demonstrates the tool's reliability and stability. Additionally, precision and recall are vital for evaluating the effectiveness of the tool's search and filtering techniques, ensuring that the tool classifies relevant documents accurately and comprehensively.

Another vital metric is the traceability of results, which is particularly important for software utilizing LLMs. It is crucial to understand why specific results were generated, providing transparency in the decision-making process. For instance, when applying

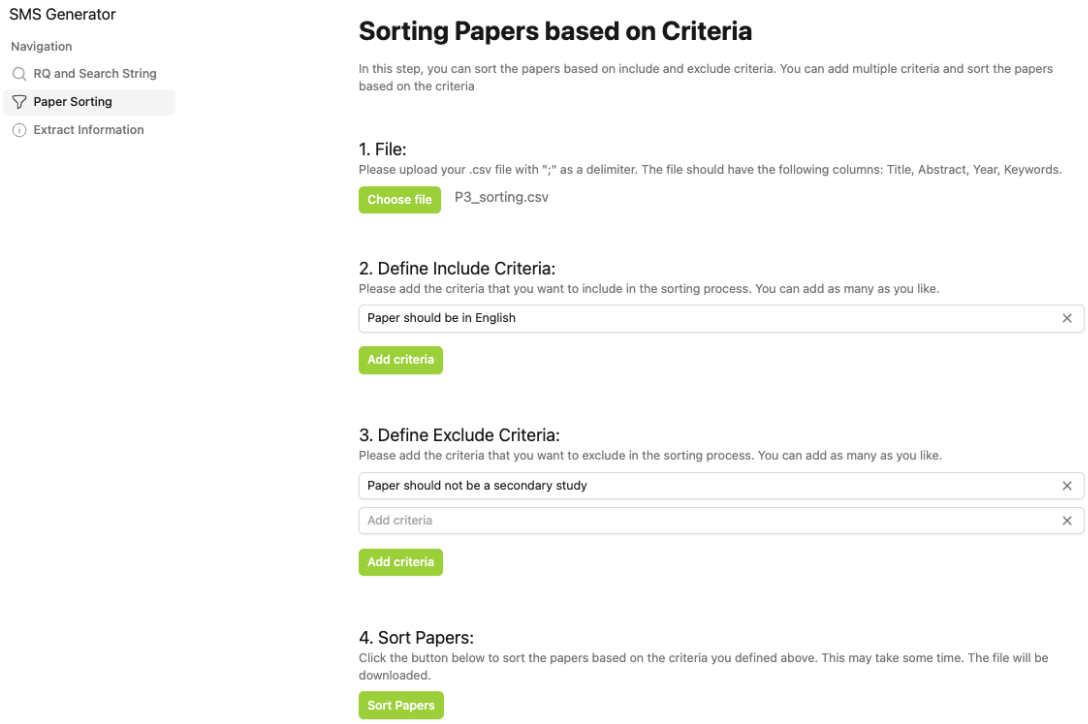


Figure 12: Sorting Page of the SMS Helper Tool

criteria to filter literature, the SMS Helper includes explanations with text references. This approach ensures that users can trace the reasoning behind each decision, fostering trust in the system's outputs and facilitating verification and validation of the results.

Robustness and error handling refer to the tool's ability to handle unexpected inputs or conditions without failing. Effective error handling mechanisms and comprehensive logging are essential for diagnosing and resolving issues, enhancing the overall reliability of the SMS Helper.

5.5 Answering RQ2

To address RQ2, the SMS Helper leverages generative AI to semi-automate various steps in SMS. The automation process begins with the creation of research questions. By inputting an objective, the SMS Helper utilizes a prompt designed to generate relevant and focused research questions. This significantly reduces the time and effort required for manual question formulation.

Next, the tool automates the construction of Boolean search strings. By analyzing the generated research questions, the SMS Helper identifies key terms and phrases, categorizes them using the PICO framework, and constructs a comprehensive Boolean query. This ensures that the literature search is both thorough and targeted, increasing the likelihood of retrieving relevant studies.

For sorting papers, the SMS Helper allows users to upload study details in a CSV format. It then applies user-defined inclusion and exclusion criteria to filter the studies, providing decisions with explanations and confidence scores. This step, which traditionally involves extensive manual review, is streamlined through the use of AI, ensuring consistency and reducing human error.

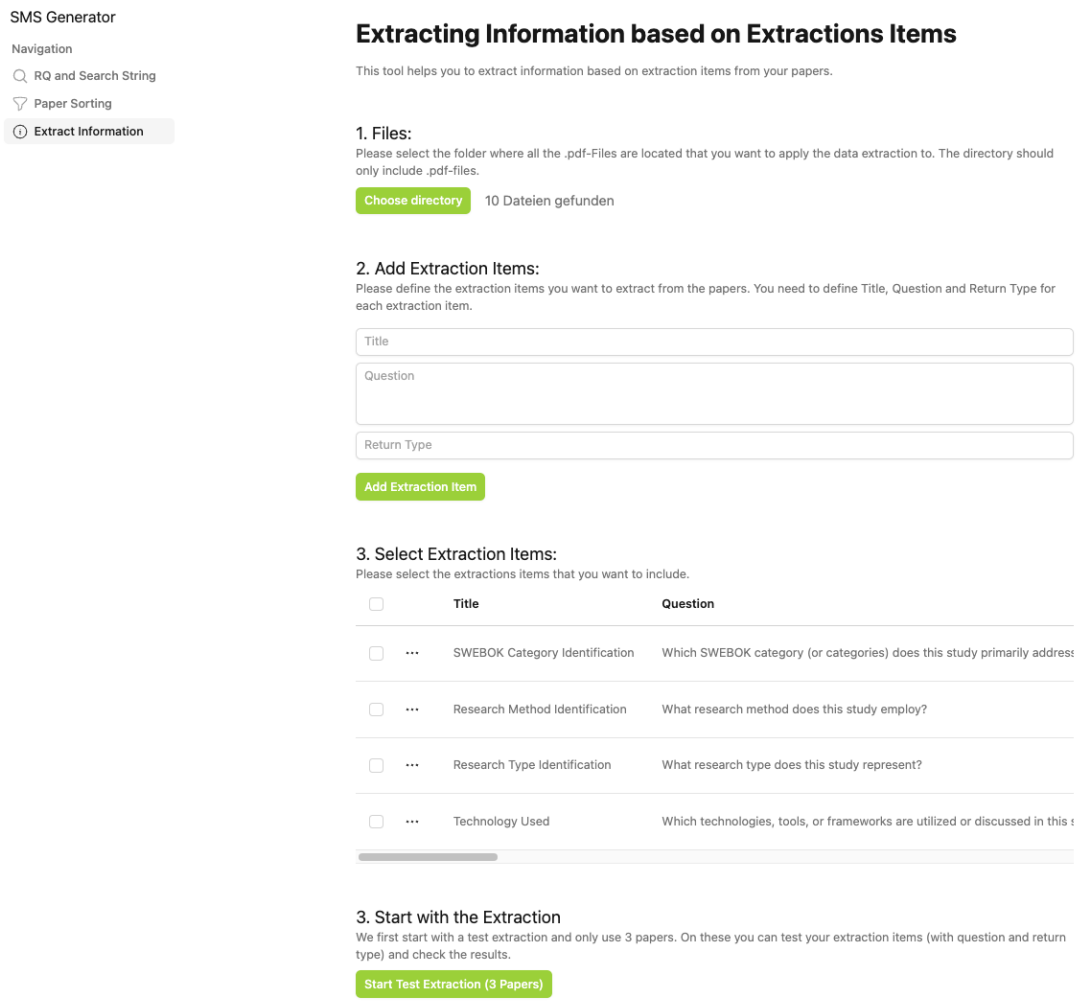


Figure 13: Extraction Page of the SMS Helper Tool

Information extraction is another critical step that is automated. Users specify the documents and extraction items, and the SMS Helper processes the files to extract relevant data, generating a structured output. This automation saves time and enhances the accuracy and consistency of the extracted information.

Additionally, the tool emphasizes the traceability of results. Each automated decision is accompanied by explanations and text references, ensuring that researchers can understand and verify the reasoning behind the tool's outputs. This transparency is crucial for maintaining trust in the system's findings.

By integrating these AI-driven functionalities, the SMS Helper effectively semi-automates the systematic mapping study process, making it more efficient and reliable while maintaining a high standard of quality and transparency.

6 Part III - Case

The purpose of this chapter is to evaluate the adjusted mapping study process in Section 4 and the AI-assisted tool developed in Section 5 by applying them to selected SMS across various domains of software engineering. This evaluation will involve re-executing the

mapping studies using the SMS Helper, comparing the results with the original studies, and analyzing the performance and effectiveness of the tool. This chapter aims to provide insights into the practical application of the AI-assisted tool, identify any discrepancies between the tool-generated and original results, and propose potential improvements for the tool and process.

6.1 Method

In this section, we detail the comprehensive methodology adopted for the development, implementation, and evaluation of the SMS Helper. The overall research process is structured to ensure a thorough examination of the tool's effectiveness across different domains of software engineering. This methodology encompasses the selection of studies, the re-execution of these studies using the SMS Helper, and the subsequent analysis of the tool's performance.

6.1.1 Selection of Studies

The selection of studies for the evaluation case was conducted with the aim of covering a broad range of SWEBOK (Software Engineering Body of Knowledge) categories. This approach ensures that the AI-assisted tool is evaluated comprehensively across different domains within software engineering, highlighting its versatility and identifying any domain-specific challenges or strengths.

The search strategy focused on identifying relevant SMS by examining the citations of the seminal paper by Petersen et al. [2015]. This approach was chosen because Petersen et al. introduced the SMS update that is widely adopted in software engineering research. By targeting studies that cite this influential work, we ensured the inclusion of high-quality and relevant SMS.

Google Scholar was used as the primary source for this search. The citation list of Petersen et al. paper was screened for relevance, and each study was first evaluated on title and then later on full text.

To ensure the selection of relevant and high-quality studies that align with the process proposed by Petersen et al. [2015], the following inclusion and exclusion criteria were established:

Inclusion Criteria:

- I1: Studies must cover Objective, Research Questions, Keywords, Search String, Inclusion- and Exclusion Criteria and Extraction Items as defined per Petersen et al. [2015].
- I2: Studies must cover one of the SWEBOK categories.
- I3: Studies must be peer-reviewed.

Exclusion Criteria:

- E1: Studies that are not written in English.
- E2: Studies that are not published in high-quality, peer-reviewed journals or conferences.
- E2: Studies that are not relevant to software engineering.

The selection process involved the following stages:

- **Initial Search:** Google Scholar was used to identify studies citing Petersen et al. [2015] paper. This search yielded a substantial number of studies.
- **Title and Abstract Screening:** The titles and abstracts of these studies were screened for relevance, resulting in a shortlist of 30 studies.
- **Full-Text Review:** A detailed review of the full texts of these shortlisted studies was conducted, and 8 primary studies that met all criteria were selected.

The selection aimed to include studies from different SWEBOK categories to ensure a comprehensive evaluation of the AI-assisted tool. This approach not only highlights the tool's applicability across various domains but also identifies any specific challenges or advantages in different areas of software engineering.

Table 1 summarizes the selected studies, the authors, their assigned SWEBOK categories, and the year of publication.

Table 1: Selected Studies

ID	**Title**	**Authors**	**SWEBOK Category**	**Year**
P1	Trends and challenges in research into the human aspects of ransomware: a systematic mapping study	Murray et al.	Software Engineering Professional Practice	2024
P2	An Analysis of MLOps Architectures: A Systematic Mapping Study	Amou Najafabadi et al.	Software Engineering Models and Methods	2024
P3	Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study	Humayun et al.	Software Quality	2020
P4	Requirements engineering for artificial intelligence systems: A systematic mapping study	Ahmad et al.	Software Requirements	2023
P5	Architecting with microservices: A systematic mapping study	Di Francesco et al.	Software Design	2019
P6	Machine Learning Applied to Software Testing: A Systematic Mapping Study	Durelli et al.	Software Testing	2019
P7	Kanban in software engineering: A systematic mapping study	Ahmad et al.	Software Engineering Process	2017
P8	Reengineering legacy applications into software product lines: a systematic mapping	Assunção et al.	Software Maintenance	2017

The selection of these studies was conducted to ensure a balanced and comprehensive evaluation of the SMS Helper across various domains of software engineering. By covering different SWEBOK categories, the evaluation case provides insights into the tool's performance in diverse contexts, highlighting its strengths and identifying areas for improvement. This approach also ensures that the evaluation is not biased towards any specific domain,

thereby providing a broad view of the tool's applicability and effectiveness in supporting SMS in software engineering.

6.1.2 Re-execution of Selected Mapping Studies

To evaluate the effectiveness of the SMS Helper and the adjusted mapping study process, we re-executed the selected SMS following a structured approach.

Search Process

The search process for the re-execution involved using only the objectives of the selected papers. We compared the RQs of each paper to ensure they matched in terms of intent and significance. Additionally, we evaluated the keywords based on their coverage. For both RQs and keywords, we calculated a Jaccard Index to measure similarity. For keywords, we only included the keywords from the SMS Helper that also were given in the original paper. Additional useful keywords were omitted.

Furthermore, we reviewed and analyzed the search strings used in the original studies. This involved a thorough examination of the search strategies to ensure they were replicable and robust. By doing this, we aimed to understand how well the search strings captured the relevant literature and how they could be optimized for use with the tool.

The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity and diversity of sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets.

Mathematically, the Jaccard Index J for two sets A and B is given by:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where $|A \cap B|$ is the number of elements in the intersection of sets A and B, and $|A \cup B|$ is the number of elements in the union of sets A and B [Real and Vargas, 1996]. The Jaccard Index ranges from 0 to 1, where 0 indicates no similarity and 1 indicates complete similarity.

Selection Process

For the Selection, we contacted the corresponding authors of the selected papers via email, requesting a complete list of all search papers and the extraction map used in their studies. If we received the full list of search papers, we applied the inclusion and exclusion criteria from the original studies to these lists and compared the results with those reported in the studies.

In cases where we did not receive a response or the requested data, we conducted our own search in literature databases using the original search strings from the studies. We constrained the search to the publication dates of the original studies to ensure consistency. From these searches, we selected 10 excluded papers (those found by the search string but not used in the studies) and 10 included papers (those that were published in the paper). The selection of these papers was done randomly, with the only criteria that they were available and relevant to the study objectives. We then applied our tool to these papers, using the same criteria as the original studies, and evaluated whether the tool correctly classified the papers.

We had to evaluate each inclusion and exclusion criterion to see if it was compatible with our tool, which only captures the title, abstract, and year of publication. As a result, criteria requiring page numbers or information on whether the paper is available or published couldn't be addressed. We therefore retained only the criteria that were applicable to our tool. Additionally, we frequently had to manually add abstracts, as they were often missing due to being a critical information point for the tool.

Extraction Process

For the extraction process, we created an extraction map using 10 randomly selected papers (from the selected paper list of the study). This map was then compared to the original extraction map from the studies, if available. We calculated a Jaccard Index to assess the similarity between our extraction map and the original. This process was contingent upon the availability of an extraction map in the original study or obtaining one from the authors.

We had to manually adapt the extraction items to be compatible with the SMS Helper. Typically, extraction items consist of one to three words, sometimes accompanied by a description. However, our tool requires each extraction item to include a title, a question, and a return type. Consequently, we had to restructure each extraction item, often necessitating the creation of new questions and return types, which naturally resulted in some deviation from the original extraction items.

6.2 Results

The results of the re-execution of the selected SMS using the tool are presented in this section. The evaluation focused on three main areas: the similarity of RQs and keywords for search, the sorting of paper and the analysis of extraction maps.

Search

To assess the alignment of the RQs and keywords between the original and re-executed studies, we calculated the Jaccard Index for each pair. The averaged Jaccard Index value for the RQs was 0.64, indicating a moderate degree of similarity. This moderate alignment can be attributed to the relatively straightforward nature of formulating RQs, which typically center around core thematic areas without significant variation in wording or scope. Consequently, the SMS Helper was able to replicate the core themes and focus areas of the original studies with a reasonable degree of accuracy.

In contrast, the keywords exhibited a lower average Jaccard Index of 0.4, reflecting more variability in the terminology used across the studies. This discrepancy can largely be attributed to the diverse search strategies employed by the original studies. For instance, studies P4 and P8 adopted a comprehensive synonym search strategy, utilizing one or two primary keywords and expanding these with 10 to 20 synonyms, as seen in figure **??**. This approach contrasts sharply with the keyword selection mechanism of the tool, which does not incorporate such extensive synonym expansion. As a result, there was a significant divergence in the keywords identified, particularly in studies P4 and P8, where the Jaccard Index for keywords was not applicable or comparably low. Table 2 provides a summary of all these values.

Table 2: Jaccard Index Values for Research Questions and Keywords

Study ID	RQs Jaccard Index	Keywords Jaccard Index
P1	0.43	0.5
P2	1	0.43
P3	0.5	0.36
P4	0.75	N/A
P5	1	0.4
P6	0.5	0.2
P7	0.6	0.5
P8	0.38	N/A

Table 3: Original and Generated Keywords

Study ID	Original Keywords	Generated Keywords
P1	ransomware, human aspect, human factor, informal aspect, social aspect, awareness, training, culture, social engineering, information security behavior, perception, attitude	ransomware, human factors, information security, social engineering, user awareness, user behavior
P2	MLOps, machine learning operations, model, pipeline, architecture, workflow, process	MLOps, pipeline, workflow
P3	Cyber, Privacy, cyber security, network security, internet security, computer security, IT security, software security, vulnerability, cyber threat, cyber Crime, challenge, risks, violence	cybersecurity, network security, IT security, cyber threats, system vulnerabilities,
P4	requirements process, requirements elicitation, requirements gathering, requirements identification, requirements analysis, requirements validation, requirements verification, requirements specification, requirements development, requirements documentation, requirements management, requirements testing, requirements driven, functional requirements, machine learning, speech recognition, deep learning, natural language processing, computer vision, machine intelligence, chatbot, chat-bot, expert systems, self driving, autonomous, recommendation system, robot, AI, ML	NOT Applicable because of different search strategy
P5	architecture, design, system, structure, microservice	microservices, architecture
P6	Test, Software Testing, Test Automation, Test Oracle, Metamorphic Test, Test Data Generation, Muation Analysis, Mutation Testing, Test Generation, Machine Learning, Support Vector Machines, Decision Trees, Learning Based	machine learning, software testing, test automation, automated test
P7	Kanban, Software	kanban
P8	feature location, concept location, concern location, feature mining, feature identification, feature mapping, reengineering, refactoring, reconstruction, migration, migrating, evolution, legacy, restructuring, re-engineering, re-structuring, application engineering, commonality, core asset, domain analysis, domain engineering, feature analysis, feature based, feature diagram, feature model, feature modeling, feature oriented, highly-configurable system, process family, product family, product line, product line engineering, software family, software product family, software product line, software reuse, SPL, variability, variability analysis, variability management, variability modeling, variability-intensive system, variant, variation, variation point	NOT Applicable because of different search strategy

Reflecting on the data in Table 3, several interesting patterns emerge. In studies P1, P2, P3, P5, P6, and P7, the generated keywords generally capture the core themes of the original keywords, but with less granularity. For example, in P1, the generated keywords "ransomware, human factors, information security, social engineering, user awareness, user behavior" cover the primary areas but lack the detailed aspects such as "training, culture, perception, attitude" found in the original list. This suggests that while the SMS Helper is effective in identifying major themes, it may miss some of the more nuanced aspects that are captured through manual keyword selection.

The study P2 shows a more pronounced difference, where the original keywords include detailed aspects like "machine learning operations, model, architecture, workflow, process," while the generated keywords are more focused, limited to "MLOps, pipeline, workflow." This indicates that the tool might simplify the scope of the search, potentially overlooking some relevant studies that would be captured by a broader set of keywords.

In study P3, the original keywords "Cyber, Privacy, cyber security, network security, internet security, computer security, IT security, software security, vulnerability, cyber threat, cyber Crime, challenge, risks, violence" reflect a wide-ranging approach to the topic, while the generated keywords are narrower, focusing on "cybersecurity, network security, IT security, cyber threats, system vulnerabilities." This narrower focus could miss important studies related to privacy, internet security, or cyber crime.

Studies P4 and P8 present a different scenario where the original search strategy involved a comprehensive expansion of primary keywords into numerous synonyms, a strategy not employed by the SMS Helper. Consequently, the tool did not generate comparable keywords for these studies, resulting in a lack of applicable data for Jaccard Index calculation. This highlights a significant limitation of the tool in handling complex keyword strategies.

These variations underscore the significant influence of search strategy on the similarity metrics, particularly the Jaccard Index, and highlight the challenges in standardizing keyword selection across diverse research methodologies. Our analysis revealed that well-written and detailed research objectives often correlated with high-quality and similar RQs, suggesting a strong link between the clarity of a study's aims and the coherence of its RQs. However, the diversity in search strategies, such as the extensive use of synonyms in studies P4 and P8, presents a notable challenge. It is evident that the breadth and depth of keyword selection can vary greatly, influenced by methodological preferences and the specific goals of each study. This diversity makes it difficult to capture the full spectrum of relevant literature with a single, standardized search strategy using our tool.

Sorting

The sorting of papers based on inclusion and exclusion criteria was evaluated using the SMS Helper. The tool was tasked with replicating the criteria-based sorting process used in the original studies to determine its effectiveness and accuracy. The analysis yielded an average Jaccard Index of 0.62, indicating a moderate level of similarity between the tool's output and the original study results. Table 4 provides insights into each Jaccard Index. To illustrate how the SMS Helper evaluates papers based on specific criteria, Table 5 presents an example output for the criterion "Primary studies on requirements engineering for AI systems." The table compares the tool's decisions, explanations, and confidence scores for included and excluded papers.

The paper titled "Integrated Description of Functional and Non-Functional Requirements for Automotive Systems Design Using SysML" was originally considered for inclusion based on its relevance to requirements engineering. The study provides a comprehensive approach to integrating functional and non-functional requirements in automotive systems using SysML, which aligns closely with the broader objectives of requirements engineering.

However, the SMS Helper tool, upon evaluation, excluded this paper due to its stringent adherence to the specified inclusion criteria. The tool's decision was based on the absence of direct references to AI systems in the abstract and keywords. Despite the paper's relevance to the field of requirements engineering, the lack of explicit mention of AI systems led to its exclusion. This instance highlights the tool's rigorous and literal application of the criteria, which ensures precision but may overlook nuanced contributions that fall within the broader scope of the research objectives.

Our tool provided a binary prediction for each paper: "include" or "exclude." For inclusion, the tool required all inclusion criteria to be met and all exclusion criteria to be absent. This stringent approach may have contributed to the moderate Jaccard Index. The rigorousness of the tool's criteria application contrasts with the possible nuances in human evaluation, where researchers might include a paper even if it does not perfectly meet all criteria, possibly due to broader interpretative flexibility.

Table 4: Jaccard Index Values for Sorting

Study ID	**Jaccard Index**
P1	0.5
P2	0.65
P3	0.6
P4	0.75
P5	0.6
P6	0.75
P7	0.7
P8	0.4

The tool's decisions were based solely on the abstract, title, and year of each paper, which sometimes led to incomplete assessments due to missing contextual information. Moreover, the tool's detailed interpretation of criteria sometimes led to exclusions that might not align with human judgments. For example, in one instance, an inclusion criterion was "Study should be empirical with a focus on cyber security vulnerabilities." The paper titled "Cyber attack modeling and simulation for network security analysis" was excluded by the tool, which reasoned that while the paper related to cybersecurity, it primarily focused on simulation modeling for intrusion detection systems, rather than directly addressing cyber security vulnerabilities empirically. This level of specificity in the tool's reasoning highlights its potential for stringent filtering, but also underscores the challenge of capturing the nuanced decision-making processes researchers might employ.

Extraction

The extraction phase of the evaluation encountered several challenges, primarily due to the unavailability of extraction maps from the original studies. Out of the eight selected studies, only two authors responded to our request for data, and only one provided the necessary extraction maps. Other than that, one paper already provided the full data in the paper information as an online link. This limited our ability to perform a comprehensive extraction analysis across all studies. The lack of response may be attributed to the age of some studies, possible data loss, or changes in the authors' employment, making it difficult to access the required information.

For the two studies where extraction was possible, we calculated Jaccard Index values of 0.48 for P2 and 0.67 for P4. These indices reflect a moderate degree of similarity between the tool-generated data and the original extraction maps. Often, the SMS Helper managed to extract similar or even more comprehensive data than the original studies, demonstrating

Table 5: Example Output for Criteria "Primary studies on requirements engineering for AI systems"

Type	Title	Decision	Explanation	Score
Included	Towards Automated Surgical Robotics: a Requirements Engineering Approach	yes	The text discusses 'a Requirements Engineering Approach' for 'Automated Surgical Robotics,' indicating a primary focus on requirements engineering specifically for AI systems in the context of surgical robotics. The goal is to 'formalize surgical requirements' and translate them into 'behavioral specifications for an autonomous robotic system,' aligning with the criteria.	0.9
Included	Integrated Description of Functional and Non-Functional Requirements for Automotive Systems Design Using SysML	no	The text focuses on requirements engineering for automotive systems using SysML, specifically targeting the integration of mechanical and electric/electronic elements. The abstract does not mention requirements engineering for AI systems. Citations: 'automotive requirements engineering', 'passenger car development', 'SysML'.	0.95
Excluded	Towards Cyber-Physical Requirement Engineering Elicitation Tool Support	no	The abstract focuses on requirement engineering for a CubeSat system and does not explicitly mention AI systems. It discusses a requirement engineering tool to improve system-wide qualities based on quality attribute scenarios, but there's no indication that this is specific to AI systems.	0.95
Excluded	Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach	no	The text focuses on architectural requirements for Cloud Computing systems, not on requirements engineering for AI systems. Relevant citations include: 'architectural features of Cloud Computing', 'requirements of end-users, enterprises that use the cloud as a platform, and cloud providers', and 'guidelines to software architects and Cloud Computing application developers'.	0.95

Table 6: Example Extraction Excerpt

Extraction Item	**Original**	**Generated**
Evaluation Methodology	Case study	Case study
Application Domain	Medical / Healthcare	Healthcare
AI Application Type	ML	Machine learning
Technical Domain	Elicitation	• Requirements engineering • Machine learning • Data analytics • Data preparation pipelines • Advanced analytics • Model evaluations • Design and development of data warehouse systems
Requirements Notation	Conceptual model	Conceptual modeling
Framework Used	GR4ML Framework	A previously developed framework for requirements elicitation and design of machine learning solutions.
Data Collection Method	Interviews / meetings	• Open-ended interviews • Observations • Voice recording sessions

its potential for effective data extraction. Sometimes it even extracted correct data for a study when the original extraction map had no data at all.

An example of the extraction outputs is provided in Table 6, comparing the original extraction items with those generated by the SMS Helper. The table highlights several key extraction items, such as "Evaluation Methodology," "Application Domain," "AI Application Type," and others. In most cases, the tool accurately replicated the original extraction items. For instance, both the original and generated outputs identified the evaluation methodology as a case study and the application domain as healthcare. The AI application type was also correctly identified as machine learning. There were instances where the SMS Helper provided more detailed outputs. For example, under the "Technical Domain" extraction item, the tool generated a comprehensive list of related topics compared to the single item "Elicitation" in the original. Similarly, the data collection methods identified by the tool were more specific. These results demonstrate the tool's ability to enhance and expand upon the original extraction items, potentially offering more in-depth insights.

However, the tool also faced significant challenges. It occasionally failed to extract relevant information accurately, possibly due to the phrasing of extraction items or complex formatting within the text. The more accurate the extraction items are outlined with questions and return types, the more quality the answer from the tool are. The tool's

performance suggests that while it can be a valuable asset in streamlining the extraction process, there is room for improvement in its ability to handle diverse document structures and extraction criteria.

To visually depict the variation and central tendency in the tool's performance, we utilized box plots for each category, as seen in figure 14. The box plots show that while the tool performs relatively consistently in formulating research questions and sorting papers, there is greater variability in keyword selection and information extraction. This variability underscores the challenges the tool faces in handling diverse keyword strategies and the nuances of data extraction, suggesting areas where further refinement and customization may be beneficial.

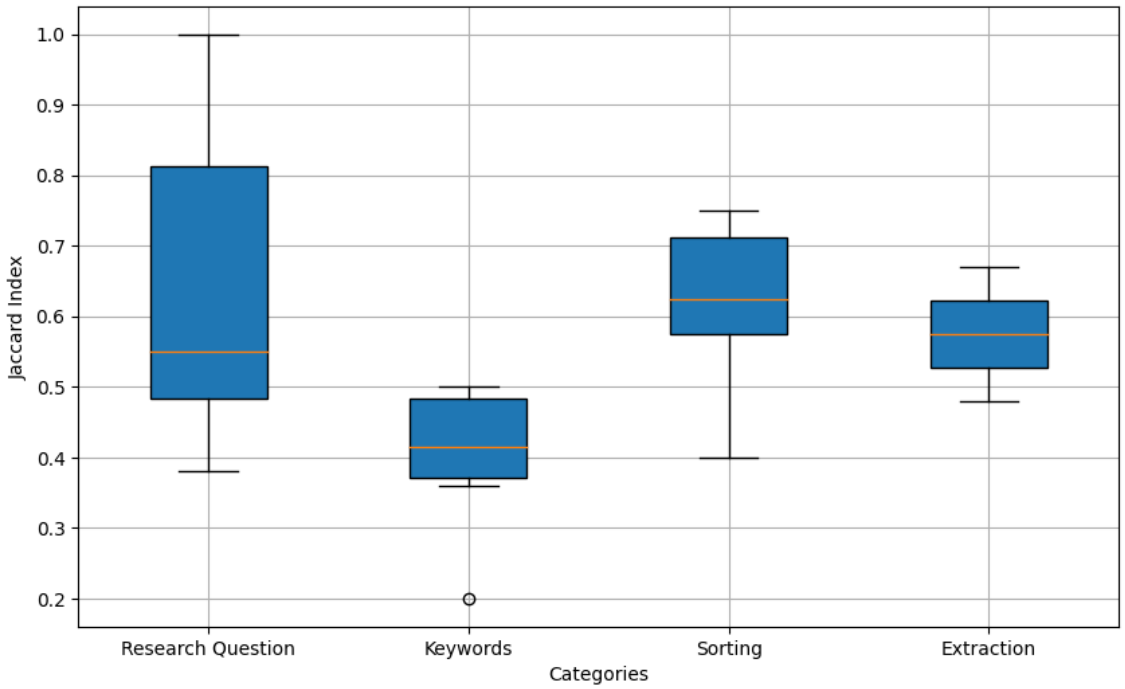


Figure 14: Boxplots for each Category

6.3 Answering RQ3

The evaluation of the SMS Helper and the adjusted mapping process was conducted through a series of case studies, which provided valuable insights into the tool's practical performance and effectiveness. The case studies involved re-executing selected systematic mapping studies across various domains of software engineering, comparing the results obtained using the SMS Helper with those from traditional manual methods.

The results indicated that the SMS Helper significantly reduced the time and effort required for needed tasks. The tool's ability to automate these tasks improved consistency and reduced the potential for human error. However, the evaluation also highlighted some limitations, particularly in the areas of keyword generation and criteria application. The tool's strict adherence to predefined criteria and text-based information sometimes led to excluding relevant studies, which might have been included by human reviewers who can interpret nuances and context.

Overall, the tool and adjusted mapping process performed effectively, demonstrating clear benefits in terms of efficiency and consistency. The findings suggest that while the SMS Helper can streamline many aspects of the systematic mapping study process, human

oversight remains essential, especially in tasks requiring nuanced judgment and context-sensitive decision-making. The case studies confirmed that the integration of AI in SMS can enhance the research process, but a balance between automation and human expertise is crucial for achieving optimal results.

7 Discussion

In this thesis, we explored the enhancement of SMS in software engineering through the integration of an AI-assisted tool, the SMS Helper. This discussion synthesizes the findings from the previous chapters, evaluating the tool's performance, the adjusted mapping study process, and the implications for future research.

The integration of the SMS Helper has significantly streamlined the SMS process. By automating critical steps such as research question generation, keyword selection, sorting of papers and information extraction, the tool has reduced the manual workload traditionally required in these studies. This automation accelerates the research process and enhances consistency and accuracy, addressing common issues such as human error and bias.

The moderate Jaccard Index scores for research questions (0.64) and keywords (0.4) indicate that while the tool can replicate core themes and focus areas, there are challenges in capturing the nuance and specificity often present in manual processes. This finding suggests that while AI can effectively support and enhance systematic mapping studies, it cannot fully replace the critical thinking and expertise of human researchers.

The variability in keyword selection strategies among different studies, as seen in the lower Jaccard Index scores for keywords, underscores a key challenge in standardizing the mapping study process. The use of extensive synonyms and domain-specific terminology in some studies highlighted the limitations of the tool's current keyword generation capabilities. This variability impacts the comprehensiveness and relevance of search strings, which are crucial for effective literature retrieval.

Additionally, the tool's approach to sorting papers, which relies heavily on titles, abstracts, and publication years, revealed limitations in fully capturing the criteria used by human reviewers. The moderate similarity in sorting results (Jaccard Index of 0.62) suggests a need for better prompts that can apply inclusion and exclusion criteria with greater nuance.

It is important to note that the AI's strictness in evaluating papers against criteria does not necessarily indicate a flaw in the tool. In numerous instances, the tool's strict adherence to the text and criteria might lead to more objective and unbiased decisions compared to human reviewers, who may be influenced by subjective factors or implicit biases. The AI's capacity to evaluate content strictly based on provided text ensures a consistent application of criteria, potentially reducing the inclusion of papers that do not fully meet the established standards. This characteristic highlights the value of AI assistance in tasks such as applying inclusion or exclusion criteria, where objectivity and consistency are paramount.

The practical evaluation of our tool in re-executing selected studies demonstrated both the tool's potential and its current limitations. While it successfully replicated certain aspects of the systematic mapping study process, such as research question formulation and data extraction, discrepancies in keyword selection and criteria application indicate areas for improvement. The feedback loop with users, providing explanations and confidence scores for automated decisions, was a valuable feature, enhancing transparency in the tool's outputs and leaving room for human decision-making.

However, the tool's reliance on available data, such as abstracts and titles, limits its ability to handle more complex criteria or extraction tasks that require deeper contextual

understanding. This limitation highlights the ongoing need for human oversight, particularly in interpreting nuanced research contexts and ensuring the accuracy of extracted information.

The findings of this thesis highlight several key areas for future research and tool enhancement. First, there is a need to refine the tool's capabilities in handling diverse keyword strategies and complex criteria. This could involve integrating more search techniques and expanding the tool's prompts to more terminology and concepts. Second, there is a need for researchers to view and edit the prompts that are being used, so the researcher has full transparency and flexibility. And lastly, researchers need to be able to get the text citations for the extraction map so they can evaluate if the found text passages are applicable to that extraction item. This once more gives more transparency to researcher.

8 Conclusion

This thesis explored the integration of an AI-assisted tool, the SMS Helper, to enhance the systematic mapping study process in software engineering. Our findings highlight the significant potential of this tool in automating key steps such as research question generation, keyword selection, paper sorting and information extraction, thereby increasing efficiency and consistency.

The use of the SMS Helper demonstrated moderate success in aligning with original study results, particularly in research question formulation and data extraction. However, variability in keyword strategies and the challenges in sorting papers according to nuanced criteria highlighted the limitations of the tool, emphasizing the continued necessity for human expertise in these processes.

Future research should focus on refining the SMS Helper's capabilities, particularly in handling diverse keyword strategies and complex inclusion and exclusion criteria. Integrating more advanced prompts could significantly enhance its accuracy and relevance across various research domains. Additionally, improving the tool's user interface and providing greater customization options will be crucial for enhancing its practical applicability and user experience.

The practical evaluation of the tool underscored its potential to streamline the systematic mapping study process, yet also revealed areas where human oversight remains essential. This balance of automation and expert input will be key in developing effective research tools that can support but not entirely replace the nuanced judgment required in academic research.

Overall, the SMS Helper represents a significant step forward in the methodology of systematic mapping studies. By enhancing efficiency and accuracy, it allows researchers to focus more on critical analysis and less on routine tasks. However, the findings of this thesis also emphasize the need for ongoing development and refinement of AI-assisted tools in research, ensuring they complement rather than replace human expertise. This work lays the foundation for future advancements in AI-assisted research methodologies, promising continued improvements in the field of software engineering research.

Conflict of interest

No confict of interest to declare.

References

Khlood Ahmad, Mohamed Abdelrazek, Chetan Arora, Muneera Bano, and John Grundy. Requirements engineering for artificial intelligence systems: A systematic mapping study. *Information and Software Technology*, 158:107176, 2023.

Muhammad Ovais Ahmad, Denis Dennehy, Kieran Conboy, and Markku Oivo. Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137:96–113, 2018.

Ahmad Alshami, Moustafa Elsayed, Eslam Ali, Abdelrahman E. E. Eltoukhy, and Tarek Zayed. Harnessing the power of chatgpt for automating systematic review process: Methodology, case study, limitations, and future directions. *Systems*, 11(7), 2023. ISSN 2079-8954. doi: 10.3390/systems11070351. URL `https://www.mdpi.com/2079-8954/11/7/351`.

Faezeh Amou Najafabadi, Justus Bogner, Ilias Gerostathopoulos, and Patricia Lago. An analysis of mlops architectures: A systematic mapping study. *arXiv e-prints*, pages arXiv–2406, 2024.

Wesley KG Assunção, Roberto E Lopez-Herrejon, Lukas Linsbauer, Silvia R Vergilio, and Alexander Egyed. Reengineering legacy applications into software product lines: a systematic mapping. *Empirical Software Engineering*, 22:2972–3016, 2017.

Vincent Claveau. Neural text generation for query expansion in information retrieval. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, WI-IAT '21, page 202–209, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391153. doi: 10.1145/3486622.3493957.

Paolo Di Francesco, Patricia Lago, and Ivano Malavolta. Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, 150:77–97, 2019.

Vinicius HS Durelli, Rafael S Durelli, Simone S Borges, Andre T Endo, Marcelo M Eler, Diego RC Dias, and Marcelo P Guimarães. Machine learning applied to software testing: A systematic mapping study. *IEEE Transactions on Reliability*, 68(3):1189–1212, 2019.

Bent Flyvbjerg. Case study. *The Sage handbook of qualitative research*, 4:301–316, 2011.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Fatih Gurcan and Setenay Sevik. Human-computer interaction: A literature analysis from 1998 to 2018 using automated text mining. In *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pages 1–5, 2019. doi: 10.1109/UBMYK48245.2019.8965599.

Mamoona Humayun, Mahmood Niazi, NZ Jhanjhi, Mohammad Alshayeb, and Sajjad Mahmood. Cyber security threats and vulnerabilities: a systematic mapping study. *Arabian Journal for Science and Engineering*, 45:3171–3189, 2020.

N. Jenko, S. Ariyaratne, L. Jeys, S. Evans, K.P. Iyengar, and R. Botchu. An evaluation of ai generated literature reviews in musculoskeletal radiology. *The Surgeon*, 2024. ISSN 1479-666X. doi: https://doi.org/10.1016/j.surge.2023.12.005. URL `https://www.sciencedirect.com/science/article/pii/S1479666X24000088`.

Shirley Kempeneer, Ali Pirannejad, and Johan Wolswinkel. Open government data from a le-

gal perspective: An ai-driven systematic literature review. *Government Information Quarterly*, 40(3):101823, 2023. ISSN 0740-624X. doi: https://doi.org/10.1016/j.giq.2023.101823. URL https://www.sciencedirect.com/science/article/pii/S0740624X23000230.

Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. 2, 01 2007.

Barbara Kitchenham, David Budgen, and Pearl Brereton. The value of mapping studies–a participant-observer case study. *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*, 01 2010.

Inc. LangChain. Q&A with RAG | Langchain — python.langchain.com. https://python.langchain.com/docs/use_cases/question_answering/. [Accessed 05-03-2024].

Hang Li, Harrisen Scells, and Guido Zuccon. Systematic review automation tools for end-to-end query formulation. SIGIR '20, page 2141–2144, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401402.

Garret Murray, Malin Falkeling, and Shang Gao. Trends and challenges in research into the human aspects of ransomware: a systematic mapping study. *Information & Computer Security*, 2024.

Henry Müller, Simran Pachnanda, Felix Pahl, and Christopher Rosenqvist. The application of artificial intelligence on different types of literature reviews - a comparative study. In *2022 International Conference on Applied Artificial Intelligence (ICAPAI)*, pages 1–7, 2022. doi: 10.1109/ICAPAI55158.2022.9801564.

Mohammad Noroozi, Hanieh Rastegar Moghaddam, Ankit Shah, Hadi Charkhgard, Sudeep Sarkar, Tapas K. Das, and Timothy Pohland. An ai-assisted systematic literature review of the impact of vehicle automation on energy consumption. *IEEE Transactions on Intelligent Vehicles*, 8(6):3572–3592, 2023. doi: 10.1109/TIV.2023.3268300.

Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *12th international conference on evaluation and assessment in software engineering (EASE)*. BCS Learning & Development, 2008.

Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2015.03.007. URL https://www.sciencedirect.com/science/article/pii/S0950584915000646.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Raimundo Real and Juan M Vargas. The probabilistic basis of jaccard's index of similarity. *Systematic biology*, 45(3):380–385, 1996.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023.

Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. Can chatgpt write a good boolean query for systematic review literature search? In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1426–1436, 2023.