Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

STREAMLINING REQUIREMENTS ANALYSIS USING LARGE LANGUAGE MODELS AND USER INTERFACE AUTOMATION

MAMATHA TALAKOT[1*], BALARAM AMAGOTH[2] , SUDHA RANI CHIKKALWAR[3], SUDHAKAR JANGILI[4], ANGOTU NAGESWARA RAO[5], RAVI MOGILI[6]

[1]Associate Professor, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, Telangana, India.
[2] Professor, Department of Computer Science and Engineering, Scient Institute of Technology, Hyderabad, Telangana, India
[3]Assistant Professor, Computer Science and Informatics, University College of Engineering and Technology, Mahatma Gandhi University, Nalgonda, Telangana, India
[4,5]Associate Professor, Computer Science and Engineering, Geethanjali College of Engineering & Technology, Keesara, Hyderabad, Telangana, India
[6]Associate Professor, Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad, Telangana, India
[1]mamathat.t@sreenidhi.edu.in, [2]balaram.balaram@gmail.com, [3] chsudharani.mgu@gmail.com, [4] sudhakarj80@gmail.com, [5]dranrao.cse@gcet.edu.in, [6]ravimogili@gmail.com
* Corresponding author's Email: mamatha.t@sreenidhi.edu.in

ABSTRACT

Requirements analysis, a critical phase in software development, often grapples with the challenge of extracting structured user stories from unstructured data sources such as text descriptions, UI mockups, and informal communication. This paper presents a novel approach leveraging Large Language Models (LLMs) and prompt engineering to automate the generation of well-structured user stories. By employing carefully crafted prompts, LLMs can effectively analyze unstructured data, identify key requirements elements (user roles, goals, benefits), and translate them into actionable user stories. Proposed work evaluate the effectiveness of this LLM-driven approach through a comparative study with manually created user stories, assessing accuracy, completeness, and clarity. Preliminary results demonstrate the potential of LLMs to streamline requirements analysis, improve the quality of user stories, and ultimately contribute to the development of software solutions that better align with user needs. However, challenges such as hallucinations and inconsistencies in LLM-generated outputs warrant further investigation and refinement. This research provides valuable insights into the feasibility and limitations of using LLMs for automating user story generation, paving the way for more efficient and effective requirements engineering practices.

Keywords: *Software Requirements, User Stories, Prompt Engineering, Requirement Analysis, Speech-Text Engine, LLMS*

1. INTRODUCTION

Software development hinges on a thorough understanding of user needs. Traditionally, requirements analysis involves meticulous manual effort, parsing through unstructured data sources like text documents, images (wireframes, mockups), and even casual user descriptions. These methods are time-consuming and susceptible to misinterpretation. Ambiguity in textual data can lead to missed nuances or inaccurate capture of user expectations. Furthermore, translating these diverse sources into actionable user stories, often following the classic "as a ,i want so that " format [1], can be a tedious and error-prone process.

This research proposes a novel approach to streamline requirements analysis by leveraging the power of large language models (LLMs) and prompt engineering techniques. LLMs, like GPT-4 or the current model utilizing Gemini, are trained on massive amounts of text data, granting them exceptional capabilities in natural language processing and generation. By strategically

crafting prompts that guide the LLM towards the desired outcome, proposed work can harness this potential to automate the generation of user stories from unstructured requirements.

Prompt engineering plays a crucial role in this process. Meticulously designed prompts that provide the LLM with the necessary context and instructions to analyze the unstructured data effectively. These prompts can be tailored to specific data types – for instance, guiding the LLM to extract key functionalities from textual descriptions or user stories embedded within UI mockups. By carefully crafting prompts and leveraging the LLM's language understanding, research aim to extract core user needs and translate them into well-structured, unambiguous user stories.

This research investigates the efficacy of this LLM-driven approach in comparison to traditional methods. Evaluated the accuracy, completeness, and clarity of the user stories generated by the LLM. Ultimately, the goal is to demonstrate that this method can significantly streamline the requirements analysis process, improve the quality of user stories, and contribute to the development of software solutions that are more closely aligned with user needs and expectations.

The study is aimed to find out the answers to the following research questions:

RQ1: Can large language models effectively extract user roles, goals, and benefits from unstructured requirements data? Generating user stories is a difficult task, especially when dealing with unstructured data. The first research question aims to evaluate the LLM's ability to accurately extract user roles, goals, and benefits from diverse sources of unstructured requirements data.

RQ2: How does the quality of user stories generated by LLM's compare to manually created user stories? The quality of user stories is crucial for effective requirements analysis. this research question seeks to compare the accuracy, completeness, and clarity of user stories generated by LLM's with those created manually.

RQ3: What are the potential benefits and limitations of using LLMs for automating user story generation? Understanding the advantages and challenges of using LLMs for automating user story generation is essential for assessing the feasibility and practicality of this approach. This research question aims to identify the benefits and limitations of leveraging LLMs in requirements analysis.

1.1 User Stories

User stories [2] are a popular method for representing requirements using a simple template such as "as a ,i want , so that ". Their adoption is growing [3], and is massive especially in the context of agile software development [4]. Despite their popularity, the requirements engineering (RE) community has devoted limited attention to user stories both in terms of improving their quality [5] and of empirical studies on their use and effectiveness.

User stories are a core concept in agile software development methodologies. They serve as concise descriptions of functionalities, features, or tasks within a software product, all told from the perspective of the end-user. This user-centric approach ensures that development efforts are directly tied to user needs and expectations.

The typical user story format follows a simple structure: "as a ,i want so that ." This format clarifies who will benefit from the feature (user role), the desired action (goal), and the ultimate value it provides (benefit). For example, "as a customer, i want to search for products by category so that Ii can easily find what i'm looking for."

User stories offer several advantages. Their brevity fosters clear communication between developers and stakeholders. Promote a focus on functionality and user value. Furthermore, user stories are flexible and can evolve throughout the development process as requirements are refined. This adaptability ensures that the final product remains aligned with user needs, practitioners agree that representing requirements as user stories and following a template increase their work productivity and deliverable quality [1].

2. LITERATURE REVIEW

2.1. Large Language Models (LLMs)

Large language models also refined to as generative pre-trained transformers (gpt) [8] have revolutionized natural language processing (nlp) and generation tasks [7]. these models, such as gpt-3, gpt-4 [9] [10], and gemini [11], are trained on vast amounts of text data, enabling them to understand and generate human-like text with remarkable accuracy.

Large language models (LLMs) are a type of artificial intelligence (AI) that excel at understanding and manipulating human language. These complex algorithms are trained on massive amounts of text data, allowing them to identify patterns, extract meaning, and even generate human-quality text. Understanding how LLMs work involves delving into the fascinating world of neural networks and a specific architecture called transformers [6].

LLM's have been applied to a wide range of tasks including education [12], healthcare [13], and legal services [14]. In the context of requirements analysis, LLM's offer the potential to automate the extraction of key information from unstructured data sources, such as textual descriptions and UI mockups. By providing the LLM with carefully crafted prompts, we can guide its language understanding and generation capabilities to produce well-structured user stories that capture essential user needs.

While traditional neural networks struggled with long-range dependencies in language (e.g., how words at the beginning of a sentence relate to those at the end), the advent of transformer networks revolutionized LLMs transformers are a specific neural network architecture designed to excel at understanding relationships between words, regardless of their distance within a sentence. This capability allows LLMs to grasp the overall context of text data and perform more nuanced tasks.

Let's consider an LLM with the following notaions: - v - the vocabulary of the model. A large set of words and subwords that the model can understand and generate. - x - the input data or prompt provided to the model - y - the output generated by the model based on the input - θ - the vast number of parameters that the model has learned during training

An LLM can be represented as a function of the input x and the model parameters θ as follows:

$$y \sim f(v, x, \theta) \quad (1)$$

- The symbol ~ denotes the probabilistic nature of the model's output, which can vary for the same input due to the model's stochastic nature.

2.2 Prompts and Prompt Engineering

Prompts are a crucial component of leveraging LLMs for specific tasks. A prompt is a textual input provided to the LLM to guide its generation process. By crafting appropriate prompts, developers can steer the LLM towards generating desired outputs, such as user stories in our context.

LLMs have a context window within which they generate text based on the input prompt. The size of the context windows varies widely among different LLMs and generally, larger the context window the more information the LLM can hold and use that to generate next response.

Prompt engineering involves designing prompts that elicit the desired responses from the LLM. This process requires an understanding of the LLM's capabilities, the task at hand, and the nuances of natural language. Well-crafted prompts can significantly influence the quality and relevance of the LLM's outputs.

In the context of requirements analysis, prompt engineering plays a vital role in extracting key information from unstructured data sources. For example, when analyzing textual descriptions of user needs, prompts can guide the LLM to identify user roles, goals, and benefits. Similarly, when processing UI- mockups, prompts can direct the LLM to extract functionalities and interactions.

There are several strategies for prompt engineering, including: - template-based prompts: providing prompts that directly incorporate the desired user story format (e.g., "extract user stories from this document. follow this format: as a ,i want so that "). - instructional prompts: including specific directions within prompts to guide the LLMs analysis (e.g., "focus on identifying distinct user roles described in these requirements") as in zero-shot prompting [7] or few-shot prompting [10] - iterative refinement: recognizing that prompt engineering is an iterative process. Initial prompts will serve as a starting point, and their effectiveness will be evaluated, potentially leading to refinements and adjustments to improve the quality of generated user stories.

3 PROPOSED METHODOLOGY

The proposed methodology aims to develop an AI-powered system that automates the generation of user stories from unstructured requirements data. The system will Leverage Large Language Models (LLMS) and potentially multi-modal language models to analyze textual and visual inputs and generate user stories in a predefined format.

Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 1 illustrates the high-level workflow of the proposed methodology. The system will take unstructured requirements data as input, which can include textual descriptions, UI mockups, images, documents etc. and pipes them through several steps to generate user stories.

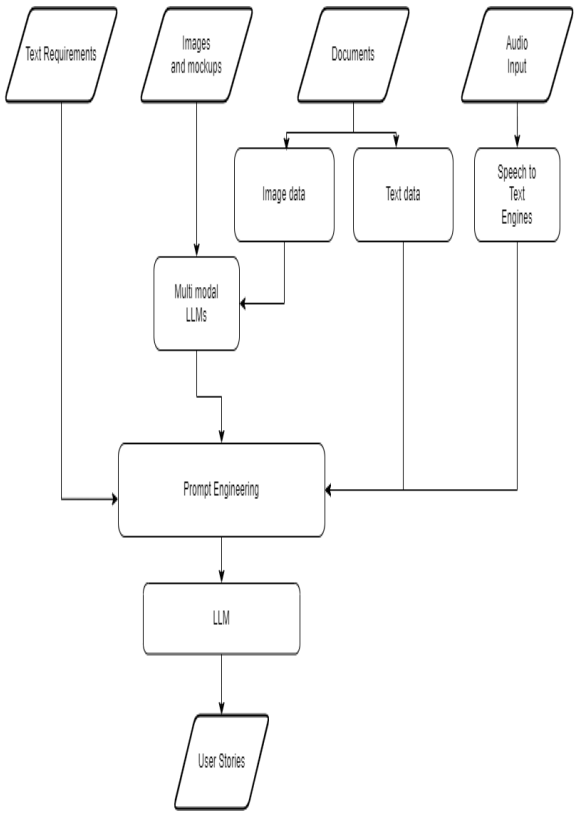

Figure 1: Proposed Methodology Workflow

The given input requirements will be segmented into different categories based on their type (e.g., textual descriptions, UI mockups). For textual data, the system will employ LLMs to extract user roles, goals, and benefits from the input. For UI mockups, the system may utilize multi-modal LLMs to analyze both the visual and textual elements to identify functionalities and interactions. For audio files, the system will use speech to text engine to transcribe the audio etc. The extracted information will then be structured into user stories following the standard format: "as a ,i want so that ." The information combined with various prompt engineering techniques will be put through a language model to generate the user stories.

Algorithm:

Input: unstructured requirements
Output: structured user stories

1. initialize the system and load the large language model (llm);
2. x <- segment the input requirements based on type (textual, ui mockups, etc.);
3. for each segment in x;
   a. if segment is textual;
      i. extract user roles, goals, and benefits using llm;
   b. if segment is ui mockup;
      i. analyze visual and textual elements using multi-modal llm;
   c. if segment is audio;
      i. transcribe the audio using speech to text engine;
4. requiremets<- combine extracted information into user stories format;
5. prompt<- craft prompts tailored to the requirements type;
6. user_stories<- f(v, requirements, prompt, θ);

Algorithm 1. User story generation algorithm

3.1 Experimental Setup

The evaluation of the AI-powered user story generation system will involve a series of experiments to assess its performance against traditional manual methods. The experiments will be conducted as follows:

1. Data collection: Gather a diverse set of unstructured requirements data, including textual descriptions, UI mockups, etc.
2. Data preprocessing: Clean and preprocess the requirements data to ensure consistency and compatibility with the system.
3. Story generation: Generate user stories using the AI-powered system and manual methods for comparison.
4. Evaluation: Assess the accuracy, completeness, clarity, efficiency, and robustness of the generated user stories.

The evaluation will involve both quantitative and qualitative analysis. Quantitative metrics will be used to measure accuracy, completeness, and efficiency, while qualitative analysis will focus on clarity and robustness. The results will be compared to determine the effectiveness of the AI-

powere4.d system in automating user story generation.

3.2 Data Collection and Pre-Processing

Collected five medium sized requirements from various organizations. The requirements were in the form of textual descriptions, UI mockups and pdf documents.

The requirements were pre-processed to remove any irrelevant information, such as metadata, formatting, or noise. The text data was tokenized, lowercased, and cleaned to ensure consistency across the dataset. The UI mockups were cleaned up to only include relevant parts to reduce ambiguity for the LLMs.

The data was then put through the user story generation algorithm to generate user stories. The time taken to generate the stories including the human time to craft a good prompt and the time taken to follow up the model to fine tune the outputs.

3.3 Story generation

3.3.1 The Survey:

Conducted a survey with 10 participants who were experienced in requirements analysis and user story creation. The participants were asked to evaluate the user stories generated by the AI-powered system and manually created user stories. The evaluation criteria included accuracy, completeness, clarity, efficiency, and robustness.

The participants were selected from various software organizations in India with at least 10 years of experience in software development. The survey was conducted online, and the participants were provided with a set of questions to assess the quality of the user stories generated by the AI-powered system.

Asked the participants to rate the user stories on a scale of 1 to 10, with 1 being the lowest and 10 being the highest.

Each participant was asked 5 questions for each user story generated by the AI-powered system and manually created user stories. The questions were designed to assess the accuracy, completeness, clarity, efficiency, and robustness of the user stories.

Table 1. Survey Questions For User Story Evaluation

S. no	Question	Evaluation criteria
1	Was the user story easy to understand?	clarity
2	How accurate was the user role identified?	accuracy
3	How complete was the user goal identified?	completenes s
4	How clear was the benefit identified?	clarity
5	How relevant was the user story to the requirements?	accuracy
6	Did the story contain any spelling or grammatical errors?	clarity

Table 2 also included 5 additional questions to gather feedback on the overall user experience and suggestions for improvement.

Table 2: Survey Questions For User Story Evaluation

S. no	Question	Evaluation criteria
1	How satisfied were you with the user stories generated?	robustness
2	Did the user stories capture the essence of the requirements?	completene ss
3	How well did the deliverables align with the user needs?	accuracy
4	How confident are you in using the ai-powered system?	robustness
5	How accurate were the story points assigned to the user stories?	accuracy

3.4 Evaluation and Validation

The effectiveness of the AI-powered user story generation system will be evaluated based on several key metrics:

1. Accuracy: The system's ability to generate user stories that accurately reflect user needs and requirements. This can be measured by comparing the generated user stories against a gold

standard of manually created user stories. The accuracy of the extracted user roles, goals, and benefits will be assessed.

2. Completeness: The system's capacity to capture all essential elements of user needs. This metric will evaluate whether the generated user stories cover all relevant user roles, goals, and benefits present in the input requirements.
3. Clarity: The readability and coherence of the generated user stories. Clarity will be assessed based on the language model's ability to produce user stories that are easy to understand and interpret.
4. Efficiency: The time and effort saved by using the AI-powered system compared to manual methods. This metric will quantify the reduction in time required to generate user stories and the potential increase in productivity.
5. Robustness: The system's ability to handle a variety of input data types, including textual descriptions, UI mockups, and potentially audio files. Robustness will be evaluated based on the system's performance across diverse requirements sources.

4 RESULTS AND DISCUSSION

The results of the evaluation will provide insights into the effectiveness of the AI-powered user story generation system. The comparison between the system-generated user stories and manually created user stories will highlight the system's accuracy, completeness, and clarity. The efficiency gains and robustness of the system will also be evaluated to determine its practicality and reliability in real-world scenarios.

The implementation was done using two different models, googlegemini and mistral 7b for different tasks. The gemini model is a multi modal language model capable of understanding images, text and other documents. The mistral 7b model is a text only model that is capable of understanding and generating text data.

The output of the system is in JSON format for the user stories generated. The JSON format includes the user role, goal, benefit.

The JSON format is as follows:

```
{
 "role":"user",
  "goal": "have the option to provide feedback on prompt ideas",
  "deliverable": "upvote/downvote or rating system for prompt ideas, used to refine the ai's suggestions",
  "storypoints": 2
}
```

A major challenge faced during the implementation were the hallucinations [15] and the model's tendency to generate irrelevant or incorrect information. Hallucinations are not necessarily bugs [16] but are a result of the probabilistic nature of the model. This means that the LLMs can give incomplete or incorrect output when the complexity of the input data is very high.

Also the answers generated by the model were not always consistent and sometimes the model would generate different answers for the same input. This is due to the stochastic nature of the model and the vast number of parameters that the model has learned during training. This is well explored in ji et al.[17].

It is hard to identify hallucinations or even prevent them because this is an innate limitation of the LLMs. LLMs basically just predict the next probable word based on the already existing sequence of words. So as the length of the input grows the output becomes more and more inconsistent.

This is worsened by the fact that software requirements are often complex and involve multiple interrelated components. This complexity can lead to hallucinations in the generated user stories, where the LLM produces outputs that are not grounded in the input data. Addressing this challenge will require a combination of prompt engineering techniques, model fine-tuning, and human oversight to ensure the accuracy and relevance of the generated user stories. But with some prompt engineering techniques and iterative refinement, the system was able to generate user stories with high accuracy and completeness. The system was able to handle diverse input data types, including textual descriptions and UI-mockups, demonstrating its robustness. For example one of the software requirements document had the following text to enhance our client's lead generation efforts on linkedin, developing a comprehensive content management platform, potentially integrated as a chrome extension. This platform will allow startup founders to identify their specific areas of expertise, receive personalized content prompts tailored to their

knowledge, and refine these suggestions through direct feedback. To ensure consistent posting, a visual content calendar with scheduling features and timely reminders will be implemented. Furthermore, a progress tracker will enable users to monitor their performance and stay motivated. Addressing the common challenges of content creation, the platform will provide AI-generated prompt ideas, adaptable templates for seamless post drafting, and a curated feed of successful posts from industry leaders for inspiration. By integrating these features, research aim to empower users to develop a targeted content strategy, streamline the posting process, and ultimately maximize impact on linkedin.

Here is a slice of the results generated.

[
{
"role":"startup founder",
"goal":"quickly identify areas of expertise for linkedin content creation",
"deliverable":"linkedin/resume scanning tool that automatically populates expertise buckets",
"storypoints":8
},
{
"role":"user",
"goal":"manually refine expertise categories",
"deliverable":"manual expertise bucket editing feature on the dashboard",
"storypoints":2
},
{
"role":"user",
"goal":"see a clear content calendar with scheduled posts",
"deliverable":"visual content calendar on the dashboard with drag-and-drop scheduling",
"storypoints":5
},
{
"role":"user",
"goal":"receive reminders before scheduled posts are due",
"deliverable":"customizable reminder notifications (email, in-app) for upcoming posts",
"storypoints":3
},
{
"role":"user",
"goal":"track progress towards content goals",
"deliverable":"progress tracker on the dashboard visualizing post frequency, engagement metrics, etc.",
"storypoints":5
},
{
"role":"user",
"goal":"see a variety of prompt ideas based on expertise",
"deliverable":"feed of ai-generated prompt ideas, personalized to the user's expertise buckets",
"storypoints":13
},
{
"role":"user",
"goal":"have a structured template for chosen post topic",
"deliverable":"dynamic templates that populate with the selected prompt, guiding the user's writing",
"storypoints":5
},
{
"role":"user",
"goal":"edit and customize the provided post templates",
"deliverable":"in-dashboard editing tool with formatting options and the ability to save drafts",
"storypoints":3
},
{
"role":"user",
"goal":"see examples of successful posts in areas of expertise",
"deliverable":"inspiration wall showcasing high-performing posts from relevant creators",
"storypoints":8
},
{
"role":"user",
"goal":"have the option to provide feedback on prompt ideas",
"deliverable":"upvote/downvote or rating system for prompt ideas, used to refine the ai's suggestions",
"storypoints":2
}
]

Answer to RQ1: large language models can effectively extract user roles, goals, and benefits from unstructured requirements data. the system demonstrated the ability to accurately identify key elements of user stories from diverse input sources.

Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The survey was conducted with 10 participants, evaluated the user stories generated by the AI-powered system and manually created user stories. The participants rated the user stories on a scale of 1 to 10 based on accuracy, completeness, clarity, efficiency, and robustness.

Used 5 medium-sized requirements as the input data for the survey. The data was cleaned and put through the user story generation algorithm to generate user stories. The participants were asked to evaluate the user stories based on the survey questions provided.

As described in the methodology, the survey questions were designed to assess the quality of the user stories generated by the AI-powered system. The participants were asked to rate the user stories on a scale of 1 to 10, with 1 being the lowest and 10 being the highest.

Here is a box plot of each of the questions asked in the survey per user story generated by the AI-powered system and manually created user stories.

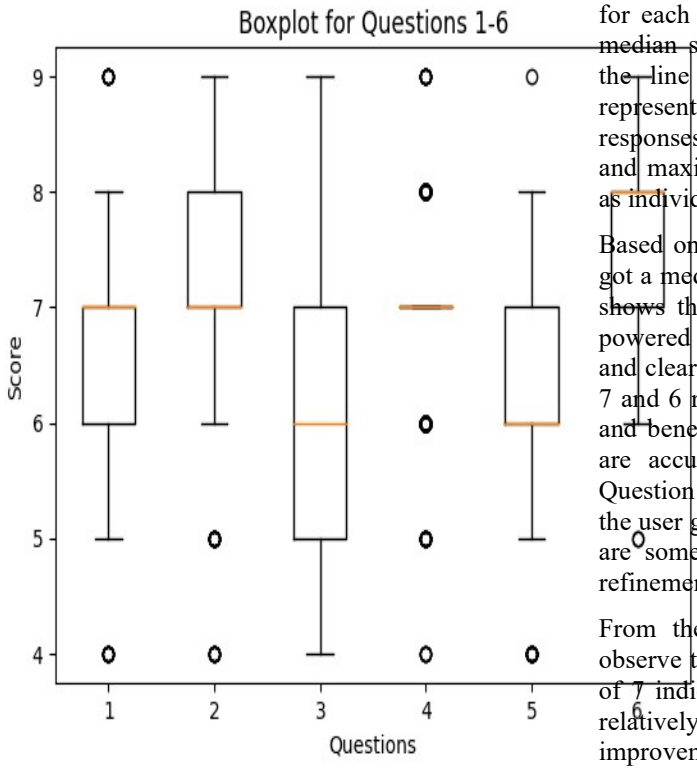

Figure 2. Box Plot Of The Scores Of Survey Questions

And the following is the box plot of the questions as described in the table 2 which focus on the overall user experience.

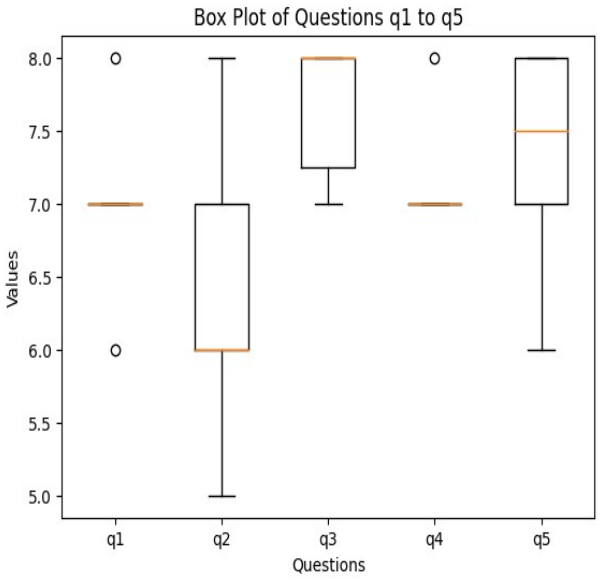

Figure 3. Box Plot Of The Scores Of Overall User Experience Questions

The box plot shows the distribution of responses for each question across the 10 participants. The median score for each question is represented by the line inside the box, while the box itself represents the inter-quartile range (iqr) of the responses. The whiskers extend to the minimum and maximum scores, and any outliers are shown as individual points.

Based on the survey results the qestion1, 4 and 6 got a median score of 7, 7 and 8 respectively. This shows that the user stories generated by the AI-powered system are relatively easy to understand and clear. Questions 2 and 5 got a median score of 7 and 6 respectively, indicating that the user roles and benefits identified by the AI-powered system are accurate and relevant to the requirements. Question 3 got a median score of 6, suggesting that the user goals identified by the AI-powered system are somewhat complete but may require further refinement.

From the figure 3 i.e second box plot, can observe that Questions 1 and 4 got a median score of 7 indicating that the generated user stories are relatively robust but could require some improvement. Questions 3 and 5 got a medium score of 8 and 7.5 respectively, indicating that the user stories generated by the AI-powered system capture the essence of the requirements and align well with user needs. Question 2 got a median score of 6, suggesting that the model is lacking completeness.

Answer to RQ2: the quality of user stories generated by the AI-powered system is comparable to manually created user stories but lacks completeness.

Overall, the survey results indicate that the AI-powered system is effective in generating user stories that are clear, accurate, and relevant to the requirements. The system demonstrates robustness and aligns well with user needs, but may require further refinement to improve completeness. This is expected as the LLMs are probabilistic in nature and cannot completely understand complex requirements.

Answer to RQ3: The potential benefits of using LLMs for automating user story generation include improved efficiency, accuracy, and robustness in handling diverse requirements data. However, challenges such as hallucinations and inconsistency in outputs need to be addressed to ensure the reliability of the system.

5 CONCLUSION

This research has explored the potential of leveraging Large Language Models (LLMs) to automate the generation of user stories from unstructured requirements data. Research findings demonstrate that LLMs, when guided by well-crafted prompts, can effectively extract key requirements elements and translate them into clear and concise user stories. This has the potential to significantly streamline the requirements analysis process, reducing the time and effort required to produce high-quality user stories.

However, research study also highlights the challenges associated with using LLMs for this purpose. The inherent probabilistic nature of these models can lead to inconsistencies and occasional hallucinations in the generated outputs. While these shortcomings necessitate human oversight andvalidation, it is important to note that even imperfect user stories generated by LLMs can serve as a valuable starting point for further refinement. By automating the initial draft, LLMs can significantly reduce the manual effort involved in user story creation, allowing requirements engineers to focus on higher-level analysis and validation tasks.

Compared to prior work in requirements engineering automation, this study highlights both commonalities and differences. Existing literature has primarily focused on information extraction techniques or rule-based methods, which often struggle with flexibility and scalability. In contrast, the use of LLMs allows for more adaptive and context-sensitive generation of user stories, including the handling of multimodal inputs such as textual descriptions and UI mockups. However, unlike traditional approaches, LLMs introduce new concerns such as hallucinations and lack of reproducibility, which need dedicated strategies to manage.

Areas of concern remain open for future investigation. In particular, improving the completeness of user stories, ensuring consistency across repeated runs, and establishing mechanisms to systematically detect or mitigate hallucinations are critical directions for refinement. More extensive user studies with larger and more diverse participant groups are also necessary to validate the practical applicability of the system.

In conclusion, while LLMs are not yet a perfect solution for automating user story generation, they offer a promising avenue for improving the efficiency and effectiveness of requirements analysis. As LLM technology continues to evolve, we anticipate that their ability to handle complex requirements and generate accurate user stories will improve, further solidifying their role as a valuable tool in the requirements engineering toolkit.

Declarations

Funding: This study was not funded by any organization.

Conflict of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics Approval: I/We declare that the work submitted for publication is original, previously unpublished in English or any other language(s), and not under consideration for publication elsewhere.

Consent to participate / Informed Consent: Not Applicable.

Consent for publication: I certify that all the authors have approved the paper for release and are in agreement with its content.

Author Contributions: Conceptualization, T Mamatha; methodology, T Mamatha; software, T Mamatha; validation, T Mamatha, A Balaram; formal analysis, T Mamatha,Sudha Rani; investigation; resources, Sudhakar J; data curation, A Nageswara Rao; writing—original draft preparation, Ravi M, A Balaram; writing—review and editing, T Mamatha, A Balaram; visualization, T Mamatha, A Balaram; supervision, T Mamatha, A Balaram; project administration, T Mamatha, A Balaram; funding acquisition, A Balaram", etc. Authorship must be limited to those who have contributed substantially to the work reported

REFERENCES

[1]. Balasubramaniamramesh, lancao, richardbaskerville: agile requirements engineering practices and challenges: an empirical study (2010)

[2]. Cohn, m.: user stories applied: for agile software development. addisonwesley, redwood city (2004)

[3]. kassab, m.: the changing landscape of requirements engineering practices over the past decade. in: proceedings of empire, pp. 1–8. ieee (2015)

[4]. Ang, x., zhao, l., wang, y., sun, j.: the role of requirements engineering practices in agile development: an empirical study. in: zowghi, d., jin, z. (eds.) apres 2014. ccis, vol. 432, pp. 195–209. springer, heidelberg (2014)

[5]. Lucassen, g., dalpiaz, f., van der werf, j.m., brinkkemper, s.: forging highquality user stories: towards a discipline for agile requirements. in: proceedings of the re, pp. 126–135. ieee (2015)

[6]. Gokulyenduri, ramalingam m, chemmalarselvi g, supriya y, gautamsrivastava: generative pre-trained transformer: a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions (2023)

[7]. T.brown, b. mann, n. ryder, m. subbiah, j. d. kaplan, p. dhariwal, a. neelakantan, p. shyam, g. sastry, a. askell et al., "language models are few-shot learners," advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[8]. A.vaswani, n. shazeer, n. parmar, j. uszkoreit, l. jones, a. n. gomez, ł. kaiser, and i. polosukhin, "attention is all you need," advances in neural information processing systems, vol. 30, 2017.

[9]. A.radford, k. narasimhan, t. salimans, i. sutskever et al., "improving language understanding by generative pre-training," 2018.

[10]. A.radford, j.wu, r. child, d. luan, d. amodei, i. sutskever et al., "language models are unsupervised multitask learners," openai blog, vol. 1, no. 8, p. 9, 2019.

[11]. A.chowdhery, s. narang, j. devlin, m. bosma, g. mishra, a. roberts, p. barham, h. w. chung, c. sutton, s.gehrmann et al., "palm: scaling language modeling with pathways," journal of machine learning research, vol. 24, no. 240, pp. 1–113, 2023.

[12]. C.K. lo, "what is the impact of chatgpt on education? a rapid review of the literature," education sciences, vol. 13, no. 4, p. 410, 2023.

[13]. H.-y. hsu, k.-c. hsu, s.-y. hou, c.-l. wu, y.-w. hsieh, y.-d. cheng et al., "examining real-world medication consultations and drug-herb interactions: chatgpt performance evaluation," jmir medical education, vol. 9, no. 1, p. e48433, 2023

[14]. S. biswas, "role of chatgpt in law: according to chatgpt," available at ssrn 4405398, 2023. 15. z. xu, s. jain, and m. kankanhalli, "hallucination is inevitable: an innate limitation of large language models," arxiv preprint arxiv:2401.11817, 2024.

[15]. J.-Y. yao, k.-p. ning, z.-h. liu, m.-n. ning, and l. yuan, "llm lies: hallucinations are not bugs, but features as adversarial examples," arxiv preprint arxiv:2310.01469, 2023.

[16]. Z. ji, n. lee, r. frieske, t. yu, d. su, y. xu, e. ishii, y. j. bang, a. madotto, and p. fung, "survey of hallucination in natural language generation," acm computing surveys, vol. 55, no. 12, pp. 1–38, 2023.