INCOSE

CHAPTER CONFERENCE

AI-Assisted Traceability Link Suggestion in MBSE for Medical Devices Using NLP

INCOSE MBSE SUMMIT · 2025

AUTHORS

Poorva Tiwari

CHAPTER

India Chapter

PUBLICATION DATE

July, 2026

SUBMISSION #

6

INCOSE PUBLICATIONS

AI-Assisted Traceability Link Suggestion in MBSE for Medical Devices using NLP

Poorva Tiwari
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu
+91 9149214955
tiwaripoorva0017@gmail.com

Abstract. It is crucial to ensure that each system requirement is accurately linked to its respective test case to maintain the safety, reliability, and regulatory compliance of medical devices. However, the manual mapping of these connections can be laborious, prone to errors, and impractical for large-scale projects. This study presents an AI-assisted framework aimed at optimizing this process by utilizing advanced semantic similarity techniques based on transformer models. We specifically employed the pre-trained all-MiniLM-L6-v2 sentence-transformer to convert both requirements and test cases into semantic embeddings, enabling precise similarity measurement and identification of optimal matches. To enhance our methodology, we also integrated the lightweight T5-small model to generate para-phrased versions of test cases, improving dataset diversity and the robustness of the matching process. Our findings indicate that the framework achieves promising performance, with a precision of 1.0, recall of 0.67, and an F1 score of 0.8. Among the 30 requirements assessed, 15 were matched with high confidence, 13 with moderate confidence, and 2 with lower confidence: importantly, every requirement was successfully matched. Although the automated method resulted in a time saving of approximately 2.5 hours compared to manual mapping, we acknowledge that traceability is often a one-time setup per project. Therefore, the primary benefit lies not just in initial time savings but in the reduction of human error, improved trace consistency, and scalability across evolving systems. It is important to note that while the framework automates the trace suggestion process, final validation by domain experts remains essential—particularly given the safety-critical nature of medical devices. The upfront effort involved in developing and integrating this framework into an Model Based SE environment is offset by the long-term gains in verification efficiency, audit readiness, and adaptability. Statistical analysis validated the efficacy of our approach, yielding a t-statistic of 21.07 and a highly significant p- value of 0.0029. The results demonstrate that integrating sentence-transformer models with data augmentation provides a scalable, efficient, and auditable solution for traceability analysis. Such innovations are especially beneficial in regulated sectors like healthcare, where accurate and transparent trace- ability is essential for certification and patient safety.

Keywords. Model-Based Systems Engineering MBSE, Medical Devices, Requirements Traceability, Test Case Automation, Artificial Intelligence (AI) in healthcare

Introduction

Background

Model-Based Systems Engineering MBSE (MODEL BASED SE)) has emerged as a revolutionary method for addressing the complexities of contemporary systems by transitioning from document-focused processes to model-oriented development. In the field of medical devices, where safety, regulatory adherence, and traceability are critical, Model Based SE provides a means to enhance the design, validation, and verification processes. Medical devices, which range from basic thermometers to intricate defibrillators and infusion pumps, necessitate stringent requirements management and thorough testing to guarantee performance and ensure patient safety. Machine Based SE facilitates improved management of these complexities by sustaining consistent, integrated models throughout the system's lifecycle.

Current Challenges

Despite the increasing implementation of Model-Based Systems Engineering MBSE (Model Based SE) in the healthcare industry, several enduring challenges persist. A significant issue is the manual traceability between system requirements and test cases, which tend to be labor-intensive, prone to errors, and inefficient. As systems become more complex and regulatory demands rise, conventional approaches to connecting requirements with verification processes are becoming less viable. Additionally, the absence of automated tools for requirements validation heightens the risk of inconsistencies, inadequate coverage, and postponed project schedules, which ultimately affects product quality and compliance preparedness.

Motivation

Driven by these challenges, this paper investigates the incorporation of Artificial Intelligence (AI) methodologies, specifically Natural Language Processing (NLP) models, within the Model-Based Systems Engineering MBSE framework to automate and improve traceability between medical device requirements and their associated test cases. The emphasis is on utilizing semantic similarity models to minimize manual labor, enhance precision, and develop a scalable solution that can be adapted for various medical devices. This research introduces a semantic matching technique where a carefully curated dataset of common medical devices, along with their standard requirements and test cases, is analyzed using advanced sentence embedding models. The proposed solution facilitates automated traceability mapping and identifies shared test scenarios across different devices, thereby revealing opportunities for reusability and standardization. By showcasing a practical AI-driven improvement to the Model Based SE process, this paper aims to address complexity through simplicity, aligning seamlessly with the theme of the summit.

Problem Statement

The creation and implementation of medical devices require strict compliance with safety, performance, and regulatory standards. Traceability, which refers to the ability to connect system requirements with their respective verification and validation processes, is a crucial element of regulatory compliance frameworks such as ISO 13485 and FDA 21 CFR Part 820. In conventional Model-Based Systems Engineering (MBSE) workflows, this traceability is established and maintained manually across various documents, spreadsheets, and system models. As the complexity of medical devices escalates, the number and detail of requirements and test cases increase significantly. Manual approaches to establishing traceability between requirements and test cases become impractical, resulting in several significant challenges:

- Time-Consuming and Labor-Intensive Process: Large systems may encompass hundreds to thousands of requirements, rendering manual linking a slow and error-prone endeavor.

- Inconsistent Coverage: Human errors can lead to overlooked requirements or incorrectly linked test cases, jeopardizing thorough validation.
- Limited Scalability: As systems develop, the manual maintenance of traceability links becomes increasingly unfeasible.
- Compliance Risks: Inaccurate or incomplete traceability may result in audit failures, regulatory noncompliance, and delays in market entry.

Despite progress in Model Based SE tools, there remains a lack of an automated, intelligent system capable of mapping requirements to test cases through semantic understanding. This paper seeks to fill this void by proposing a methodology that utilizes AI-driven semantic similarity models to automate and improve traceability between medical device requirements and their associated test cases.

Literature Survey

The utilization of Model-Based Systems Engineering (MBSE) in the development of medical devices has gained significant traction due to its effectiveness in addressing the inherent complexities of these systems. This section reviews existing literature pertaining to Model Based SE traceability, AI applications in medical device development, and pertinent regulatory frameworks.

a. Estefan (2007) presents a comprehensive overview of MBSE (Model Based SE) methodologies and their relevance across various industries, emphasizing how MBSE (Model Based SE) practices facilitate system design, development, and maintenance. The paper discusses the transition from document-centric engineering to model-centric methodologies and examines several MBSE (Model Based SE) tools, including SysML, utilized in the design and management of complex systems. This study offers essential insights into MBSE (Model Based SE) methodologies, which are vital for comprehending their application in medical devices.

b. Wang, Li, and Zhang (2020) address the challenges associated with traceability in MBSE (Model Based SE) and propose an innovative method to automate traceability processes. Their research highlights the importance of traceability in ensuring that each requirement is accurately linked to its corresponding design, testing, and verification artifacts, which is especially critical in the medical device sector where compliance with regulatory standards is paramount. Their suggested method automates these processes, minimizing manual errors and enhancing efficiency in system development.

c. Reimers and Gurevych (2019) present Sentence-BERT, a deep learning approach for generating semantically rich sentence embeddings, which can be utilized to refine the processes of requirement analysis and documentation in MBSE (Model Based SE). Their methodology im- proves natural language processing (NLP) capabilities in interpreting and aligning intricate textual requirements with the corresponding system models. Their approach enhances natural language processing (NLP) capabilities in interpreting and aligning complex textual requirements to the corresponding system models. This technique can be employed to automate the extraction and analysis of requirements in medical device development, reducing the manual effort re- quired for requirement management.

d. Doshi, Smith, and Patel (2019) examine the influence of machine learning (ML) on regulatory compliance within the context of medical device development. They propose that ML can facilitate the automation of the compliance verification process by forecasting potential compliance challenges early in the design phase. This anticipatory strategy aids in recognizing possible risks and areas of non-compliance, thus enhancing the overall efficiency of the regulatory process and shortening the time required to bring medical devices to market.

e. Blanchard and Fabrycky (2011) outline essential principles and practices in systems engineering, serving as a thorough reference for comprehending the integration of

Model Based Systems Engineering (MBSE) into wider systems engineering workflows. Their research underscores the significance of early design and requirements analysis, which is in line with the practices needed for creating medical devices that fulfill regulatory standards. Their publication offers a comprehensive overview of systems engineering principles, establishing a foundation for the application of MBSE (Model Based SE) practices in the medical device sector.

f. Kossiakoff and Sweet (2011) delve into systems engineering principles and practices, broadening the understanding of MBSE (Model Based SE) methodologies. They concentrate on how MBSE (Model Based SE) can enhance lifecycle management of intricate systems and devices, positioning it as an optimal methodology for medical device development. Their research also highlights the necessity for collaborative teamwork and integrated processes, which are vital in the multidisciplinary realm of medical device engineering. Jackson and Milne (2018) explore the effects of MBSE (Model Based SE) on the lifecycle of medical devices, particularly regarding regulatory compliance and system integration. Through an in-depth case study, they illustrate the advantages of implementing MBSE (Model Based SE) in medical device development, such as enhanced documentation, improved requirements management, and more effective verification and validation processes. Their conclusions highlight the tangible benefits of MBSE (Model Based SE) in tackling the challenges associated with regulatory requirements in the medical field.

g. Lambert and Ziegler (2020) analyze the prospects and obstacles associated with the implementation of Model-Based Systems Engineering (MBSE) for regulatory compliance in the medical device industry. They offer an in-depth examination of the intricacies present in regulatory frameworks and suggest methods for incorporating MBSE (Model Based SE) tools and techniques into compliance processes. Their research underscores the importance of a comprehensive strategy to guarantee that every facet of device development, from initial design to post- market surveillance, complies with rigorous regulatory requirements.

h. Eisenhart and McFarland (2017) investigate advanced systems engineering methodologies specifically tailored to the medical device sector. Their study emphasizes how MBSE (Model Based SE) can effectively tackle significant challenges such as system integration, risk management, and product validation. They promote the expanded application of MBSE (Model Based SE) to enhance decision-making throughout the development and lifecycle management of medical devices, particularly in relation to ensuring safety and efficacy.

i. Herzog and Janssen (2021) assess the impact of machine learning on predictive maintenance for medical devices. They highlight the potential of machine learning models to improve system reliability and mitigate the risk of device failure by forecasting maintenance requirements prior to the emergence of problems. Their findings illustrate how the integration of MBSE (Model Based SE) with machine learning can refine maintenance schedules and enhance the performance and longevity of medical devices, which is crucial for maintaining ongoing compliance with regulatory standards.

Contribution of This Paper

This study enhances the current literature by introducing a framework based on semantic similarity to automate the traceability between requirements for medical devices and their corresponding test cases. In contrast to previous methods, this framework combines AI-powered natural language processing techniques with medical device regulations and standards. Additionally, it features a visual output that enhances the clarity of traceability results, allowing stakeholders to more intuitively evaluate and confirm the mappings between requirements and test cases. The framework is designed to improve traceability while also providing reusable test cases for standard device requirements, thereby minimizing overall testing efforts and improving compliance.

Proposed Methodology

In the realm of systems engineering, particularly within the domain of medical devices, ensuring that requirements are accurately linked to their corresponding test cases is a critical yet labor-intensive task. This process, known as requirement traceability, is essential for verifying that all system requirements are adequately tested and validated. To address this challenge, we propose an innovative approach that leverages machine learning, natural language processing (NLP), and semantic similarity analysis to automate requirement traceability. The following sections in Figure 1 describe the methodology in detail, highlighting each step of the process.

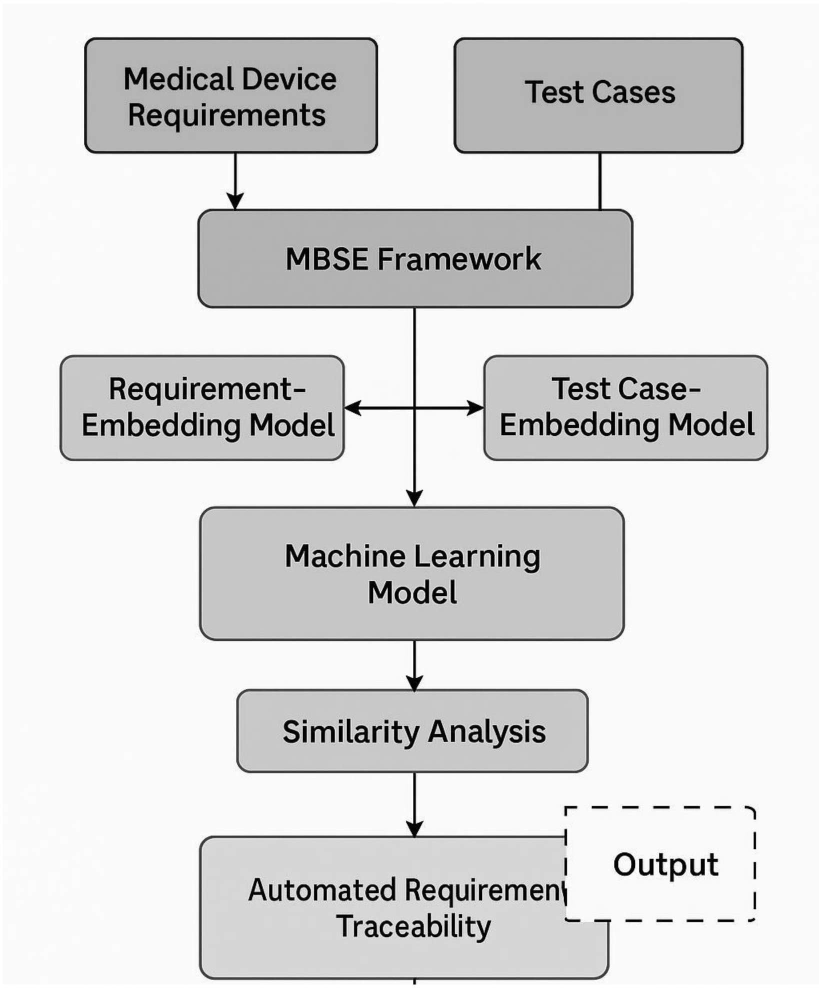

Figure 1. Proposed Methodology

Data Preparation: Establishing the Foundation

The initial phase entails the compilation of a detailed dataset that includes the requirements for medical devices along with their corresponding test cases. This dataset forms the cornerstone of the entire procedure. Each medical device is characterized by a collection of representative requirements that outline the expected functional and non-functional attributes of the device. In turn, each requirement is linked to one or more test cases aimed at verifying its adherence.

For example, a requirement for a thermometer may stipulate accuracy within a specified temperature range, while the related test case would assess the device's performance under controlled conditions.

To enhance robustness, the dataset encompasses a wide array of medical devices, including thermometers, ECG machines, and infusion pumps, each with three representative requirements and their associated test cases. This variety ensures that the model can effectively generalize across different types of devices and their requirements.

.

Textual Representation: Transforming Text into Numerical Embeddings

After the dataset has been prepared, the textual elements of both requirements and test cases are converted into numerical formats through sophisticated NLP methodologies. We utilize Sentence-BERT (SBERT), an advanced sentence embedding model, to transform the textual information into high-dimensional vector embeddings. SBERT is selected for its proficiency in capturing the semantic essence of sentences, which facilitates quantitative comparisons based on meaning rather than mere word correspondence. In this phase, each requirement and test case are encoded into a compact vector representation. These embeddings maintain the semantic connections between the texts, allowing for the assessment of how closely related two text segments are in terms of their meanings. This transformation is essential as it connects human-readable text with machine-readable data, thereby enabling the use of mathematical and statistical techniques to analyze the relationships between requirements and test cases.

Semantic Similarity Analysis: Identifying Matches

Having transformed the textual data into numerical embeddings, the subsequent step is to calculate the semantic similarity between the requirement embeddings and the test case embeddings. The primary metric employed for this calculation is cosine similarity, which assesses the cosine of the angle between two vectors. This metric yields a value ranging from -1 to 1, where a value of 1 signifies identical vectors (indicating perfect similarity), 0 denotes orthogonality (indicating no similarity), and -1 represents opposite directions (indicating dissimilarity). A predefined similarity threshold, such as 0.7, is utilized to identify strong matches between requirements and test cases. Any pair that exceeds this threshold in similarity score is deemed a valid match. This threshold is crucial for ensuring that only semantically aligned pairs are recognized, thereby minimizing noise and enhancing the accuracy of traceability links. By utilizing semantic similarity, this approach transcends mere keyword matching, effectively capturing more profound contextual relationships between requirements and test cases.

Visualization and Output Presentation

The corresponding pairs of requirements and test cases are subsequently displayed in an interactive and visually enhanced manner. This presentation is crafted to be user-friendly and intuitive, allowing stakeholders to easily interpret and validate the information. Each requirement is shown once, with all corresponding test cases organized beneath it. To improve readability and emphasize the strength of the matches, color-coding is utilized for the similarity scores:

- Green denotes strong matches (high similarity scores), reflecting a significant level of semantic alignment.
- Yellow indicates moderate matches, highlighting areas that may require further examination.
- Black text is employed within the colored cells to maintain clarity and readability, even with the application of background colors.

This visual representation not only facilitates the understanding of traceability links but also aids in decision-making by offering a clear overview of the connections between requirements and test cases.

Identification of Common Test Cases

A collection of test cases that consistently align with various device requirements has been extracted from the dataset of medical devices. This set of common test cases creates a reusable test repository that can be utilized in future projects to improve verification efficiency. By pinpointing these reusable test cases, the process minimizes redundancy and optimizes the testing phase for new medical devices, guaranteeing the consistent application of established and validated testing procedures.

Integration with Model Based SE Framework

The complete procedure is integrated into a Machine-Based Systems Engineering framework, offering a systematic approach to systems engineering tasks. This framework guarantees that the automated traceability process is in harmony with overarching systems engineering practices, fostering uniformity and scalability. The incorporation of machine learning models and semantic embeddings enhances conventional MBSE (Model Based SE) tools and methodologies, improving their functionality by auto- mating intricate tasks and minimizing manual labor.

Machine Learning Model and Similarity Analysis

Central to the process is a machine learning model that analyzes the embeddings produced by the requirement-embedding and test case-embedding models. This model conducts similarity assessments, evaluating the embeddings to find matches according to a set threshold. The machine learning aspect is essential for expanding the process to accommodate large datasets while maintaining efficiency and accuracy in the analysis. By utilizing machine learning, the system is capable of evolving and enhancing its performance over time, gaining insights from new data and improving its capacity to establish significant traceability connections.

Output: Automated Requirement Traceability

The ultimate result of the procedure is a detailed, automated requirement traceability map. This map creates explicit connections between requirements and their associated test cases, guaranteeing that every requirement is sufficiently addressed by pertinent testing protocols. The interactive and visually improved display of the outcomes allows for straightforward examination and confirmation by stake- holders, while the recognition of reusable test cases improves the overall effectiveness of the verification process.

Algorithm and Code Overview

Code Implementation and Explanation. To automate the mapping between requirements and corresponding test cases, a combination of deep learning-based semantic similarity models and traditional data handling techniques was employed. Initially, necessary Python packages such as sentence- transformers, pandas, and scikit-learn were installed. The sentence-transformers library was used to access a pre-trained transformer model (all-MiniLM-L6-v2) capable of encoding sentences into dense vector representations.

The steps of the implementation are as follows:

1. **Data Preparation:**
   To facilitate the automation of mapping requirements to their respective test cases, a blend of deep learning-based semantic similarity models and conventional data processing methods was utilized. Initially, essential Python libraries, including sentence-transformers, pandas, and

scikit-learn, were installed. The sentence-transformers library provided access to a pre-trained transformer model (all-MiniLM-L6-v2) that is adept at encoding sentences into dense vector representations. The implementation process is outlined as follows:

- Data Preparation: Requirements and test cases were either defined manually or imported from an Excel file, with any missing entries removed to maintain data integrity.
- Model Loading: The all-MiniLM-L6-v2 model was loaded, which is specifically optimized for generating embeddings that encapsulate semantic meaning, thus making it ideal for comparing requirement and test case statements.
- Sentence Embedding: Each requirement and test case was converted into its respective embedding vector using the loaded model.
- Similarity Calculation: Cosine similarity was calculated between each requirement embedding and all test case embeddings to measure their semantic proximity, resulting in a similarity matrix.
- Top Match Extraction: For each requirement, the top-N most similar test cases were determined based on the cosine similarity scores, employing a threshold-based color coding system (green for high similarity, yellow for moderate, and red for low) for enhanced visual interpretation.
- Result Presentation: The findings were organized in a structured Excel sheet, featuring columns for Requirement, Suggested Test Case, and Similarity Score. Addition- ally, advanced visualization techniques, such as horizontal bar plots, were optionally employed to graphically represent the quality of matches.

Evaluation Metrices:

a) Precision: Precision is the ratio of true positive predictions to the total number of positive predictions made (both true positives and false positives).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Precision helps in evaluating the accuracy of positive predictions. A high precision means that when the model predicts a positive match, it is usually correct, which is important in situations where false positives have a high cost (e.g., misclassifying a non-matching test case as a match).

b) Recall: Recall (also known as Sensitivity or True Positive Rate) is the ratio of true positive predictions to the total number of actual positives (the sum of true positives and false negatives).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Recall is important for measuring the model's ability to capture all relevant instances. A high recall means the model is good at identifying positive cases, even at the risk of some false positives. It is crucial when missing a true positive could have significant consequences, such as in medical diagnostics.

c) **F1 Score:** The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both the concerns of precision and recall. It ranges from 0 to 1, where 1 indicates perfect precision and recall, and 0 indicates the worst performance.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score is a useful metric when you need to balance precision and recall, especially when the class distribution is imbalanced (e.g., when there are more negative cases than positive ones). It is helpful when you care about both false positives and false negatives and need a single value to assess the model's performance.

d) **T-statisti**c: The T-statistic is a ratio that compares the difference between the observed sample mean and the hypothesized population mean relative to the variation in the sample. It is used in hypothesis testing to determine if there is a statistically significant difference between two sets of data. The formula for the T-statistic is:

$$T = \frac{\text{Sample Mean} - \text{Population Mean}}{\text{Standard Error of the Mean}}$$

The T-statistic helps in determining whether the differences observed in the data are statistically significant or if they are likely due to chance. A higher T-statistic indicates a greater likelihood that the observed difference is real, and not a result of random variation.

e) **P-value:** The P-value is the probability that the observed data (or something more extreme) would occur if the null hypothesis were true. A small P-value (typically less than 0.05) indicates strong evidence against the null hypothesis, suggesting that the observed effect is statistically significant. The P-value helps in making decisions about the validity of hypotheses. If the P-value is below a predefined threshold (often 0.05), you reject the null hypothesis, suggesting that the findings are statistically significant and not due to random chance. It helps in assessing the reliability of the results and drawing conclusions from the data.

Additionally, the system supports paraphrasing test cases using a lightweight T5 transformer model to enhance dataset variety when needed. This approach allows for rapid, scalable, and explainable traceability analysis between requirements and test cases in medical device projects.

Output Explanation. Figure 2 shows a structured table is created listing different medical devices, their model-based systems engineering (MBSE) use cases, and associated typical requirements, which serve as input for the traceability matching.

This table forms the initial dataset of requirements that will later be compared against test cases

Device	MBSE Use Case	Typical Requirements
Thermometers	Sensor calibration, real-time measurement	1. Must measure temperature within ±0.1°C accuracy. 2. Should have an auto-off feature to conserve battery. 3. Must have a range of 32°C to 43°C for human use.
Blood Pressure Monitors	Signal integrity, safety	1. Should operate within 0–300 mmHg range. 2. Must comply with IEC 80601-2-30 standards. 3. Measurement accuracy within ±5 mmHg.
Pulse Oximeters	Signal processing under motion	1. Accuracy within ±2% for SpO_2 between 70–100%. 2. Must update reading every 1 second. 3. Must perform well under motion and varying light conditions.
Glucometers	Data precision, memory logging	1. Must log a minimum of 100 readings. 2. Should have an error margin of <15% compared to lab test. 3. Must store data for at least 30 days.

Figure 2. Device Requirements Table

Figure 3 illustrates the process by which the system downloads and installs the all-MiniLM-L6-v2 sentence-transformer model, intended for transforming requirements and test cases into vector embeddings for the purpose of similarity assessment. The green progress bars indicate the successful loading of the model components required for generating embeddings.

The all-MiniLM-L6-v2 model is a compact, high-efficiency sentence-transformer designed for tasks involving semantic similarity. It utilizes the Transformer architecture to condense large language models into a significantly smaller format while maintaining accuracy. This model specifically converts sentences and brief texts into dense 384-dimensional vector embeddings that encapsulate their semantic significance. These embeddings can be efficiently compared using cosine similarity to evaluate the degree of relatedness between two sentences. Owing to its optimal combination of speed, minimal memory consumption, and robust performance, the all-MiniLM-L6-v2 model is especially well-suited for real-time applications such as matching requirements with test cases, where rapid and precise semantic comprehension is crucial.

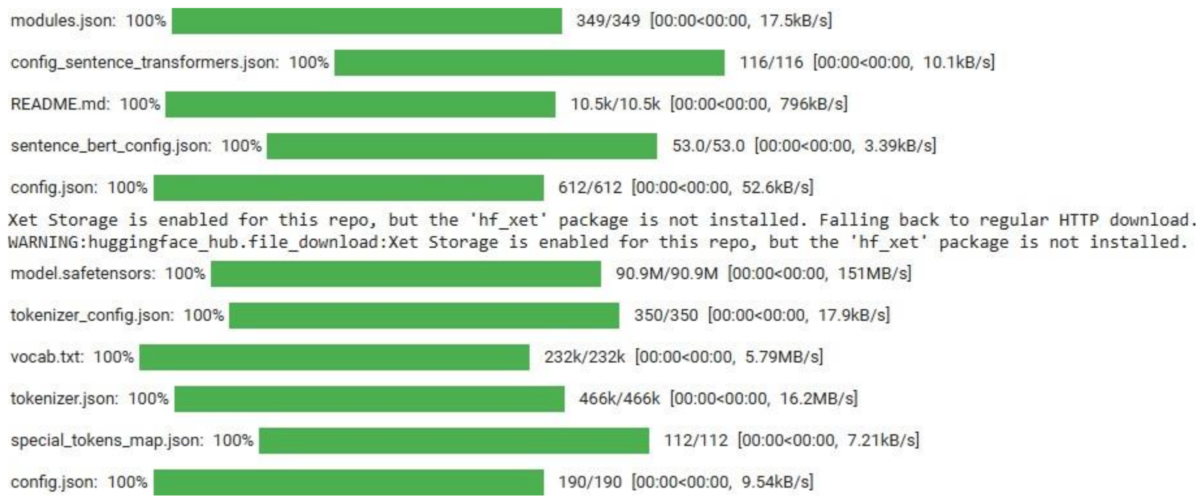

Figure 3. Model Download Progress - Sentence Transformer

Figure 4 illustrates the lightweight T5-small model, whose tokenizer is utilized to rephrase original test cases, facilitating the creation of various synthetic test case variations. This process improves the quality and robustness of the dataset for subsequent semantic matching. The T5-small model is a streamlined version of the Text-to-Text Transfer Transformer (T5) created by Google. It approaches all natural language processing (NLP) tasks—such as translation, summarization, and paraphrasing—as a text-to-text challenge. The T5-small model preserves the essential functionalities of the larger T5 architecture but with a significantly reduced number of parameters, resulting in enhanced speed and memory efficiency while still delivering impressive performance. In this study, T5-small is employed to paraphrase test cases, producing multiple linguistic variations of a given sentence to increase the dataset's diversity.

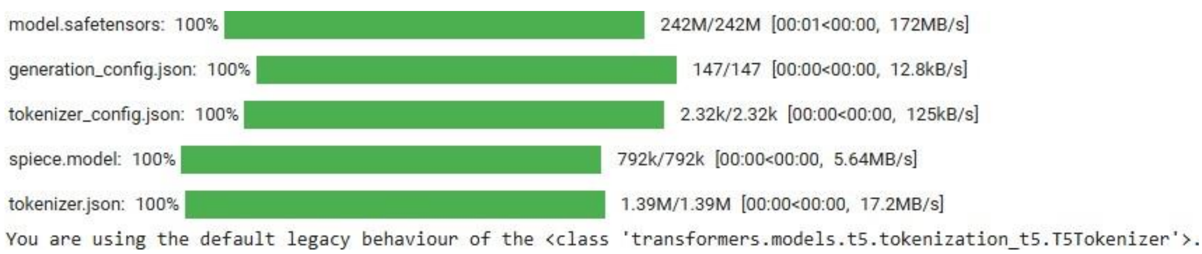

Figure 4. Model Download Progress - T5 Model for Paraphrasing

In figure 5, each requirement is matched to its top three most semantically similar test cases, and the corresponding similarity scores are displayed for interpretability. This output helps in quickly identifying the best test cases for validating each requirement based on their language similarity.

```
📌 Requirement [1]: The device must monitor ECG signals continuously for at least 24 hours.
  1. Test if the device can continuously monitor ECG signals for 24 hours without interruption.
     ➤ Similarity: 0.88
  2. Test whether the device stores ECG data correctly for a duration of 7 days.
     ➤ Similarity: 0.76
  3. Verify if the ECG monitor triggers an alert for abnormal waveforms.
     ➤ Similarity: 0.59

📌 Requirement [2]: The defibrillator shall deliver shocks with a voltage range between 150V and 300V.
  1. Verify if the defibrillator delivers shocks within the specified voltage range (150V - 300V).
     ➤ Similarity: 0.86
  2. Check if the defibrillator automatically selects the correct shock setting for adult and pediatric patients.
     ➤ Similarity: 0.55
  3. Verify if the defibrillator charges up to 80% in under 5 minutes.
     ➤ Similarity: 0.49

📌 Requirement [3]: The device should measure the oxygen saturation (SpO2) within 95-100%.
  1. Test whether the device measures SpO2 accurately between 95% and 100%.
     ➤ Similarity: 0.78
  2. Check if the pulse oximeter displays the SpO2 value within 5 seconds.
     ➤ Similarity: 0.67
  3. Test if the SpO2 device provides readings within 2% of a known reference.
     ➤ Similarity: 0.65
```

Figure 5. Requirement-Test Case Matching Results

Figure 6 illustrates the previously mentioned data in a manner that is both visually clear and easy to understand. The figure utilizes a color-coded system to represent the similarity scores between test cases and requirements, thereby enhancing the clarity and immediacy of the information presented. Each similarity score, displayed in the final column, serves a dual function: it provides an exact numerical value that indicates the level of correspondence, while also offering a visual indicator through color coding, enabling users to quickly evaluate the quality of the match without relying solely on numerical data. The color-coding adheres to a straightforward and logical framework:

- Green signifies a similarity score exceeding 80%, indicating a strong correlation between the test case and the requirement.
- Yellow denotes a score ranging from 60% to 80%, indicating moderate similarity where some discrepancies may be present.
- Red is assigned to any score below 60%, clearly indicating a substantial mismatch that necessitates further investigation.

This visual strategy not only simplifies the identification of well-aligned and misaligned test cases at a glance but also enhances overall readability, making the analysis more accessible and efficient for stakeholders.

	Requirement	Matched Test Case	Similarity Score
0	The device must monitor ECG signals continuously for at least 24 hours.	Test if the device can continuously monitor ECG signals for 24 hours without interruption.	0.880541
1		Test whether the device stores ECG data correctly for a duration of 7 days.	0.757348
2		Verify if the ECG monitor triggers an alert for abnormal waveforms.	0.594375
3	The defibrillator shall deliver shocks with a voltage range between 150V and 300V.	Verify if the defibrillator delivers shocks within the specified voltage range (150V - 300V).	0.859710
4		Check if the defibrillator automatically selects the correct shock setting for adult and pediatric patients.	0.549349
5		Verify if the defibrillator charges up to 80% in under 5 minutes.	0.491774
6	The device should measure the oxygen saturation (SpO2) within 95-100%.	Test whether the device measures SpO2 accurately between 95% and 100%.	0.779530
7		Check if the pulse oximeter displays the SpO2 value within 5 seconds.	0.671570
8		Test if the SpO2 device provides readings within 2% of a known reference.	0.649160
9	The heart rate monitor must support a range of 30-220 beats per minute (bpm).	Verify if the heart rate monitor reads accurately between 30 bpm and 220 bpm.	0.779691
10		Test the blood pressure monitor for accuracy within ±5 mmHg.	0.498593
11		Check whether the pulse oximeter provides accurate pulse rate readings.	0.491206
12	The device must be able to communicate wirelessly with a mobile application via Bluetooth.	Test if the device can establish a stable Bluetooth connection with the mobile app.	0.697913
13		Verify if the device can connect to a cloud system and upload data correctly.	0.429430
14		Verify if the device uses encryption for all patient data and meets HIPAA standards.	0.297147

Figure 6. Evaluation of the Requirement-Test Case Matching

The sub-figures presented in Figure 7 illustrate the graphical depiction of the similarity scores referenced in the preceding table. This bar chart highlights the top three test cases that fulfill the requirement: 'The device must continuously monitor ECG signals for a minimum of 24 hours.' The similarity scores reflect the degree to which each test case corresponds with the requirement. The test case with the highest score is 'Test if the device can continuously monitor ECG signals for 24 hours without interruption' (score: 0.89), followed by 'Test whether the device accurately stores ECG data for a period of 7 days' (score: 0.76), and 'Verify if the ECG monitor activates an alert for abnormal waveforms' (score: 0.61). This visualization aids in determining which tests most effectively validate the continuous monitoring requirement.

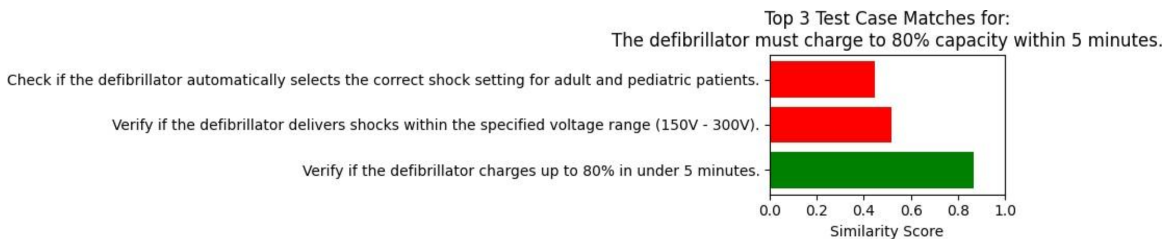

Figure 7. Similarity Score (Graphical Representation)

Figure 8 provides a detailed assessment of the alignment between test case matches and the specified requirements, utilizing essential performance metrics alongside supplementary statistical data. A precision score of 1 signifies that every identified high-quality match was a true positive, with no instances of false positives. The recall rate of 0.67 indicates that 67% of all potential true positive cases were accurately recognized, suggesting opportunities for improvement in capturing all pertinent matches. The F1 Score of 0.8 demonstrates a well-balanced performance between precision and recall, reflecting a commendable overall match quality. The categorization of matches is also illustrated: 15 out of 30 high-quality matches indicate a strong correspondence between test cases and requirements, while 13 out of 30 moderate matches suggest a reasonable fit with potential for enhancement, and 2 out of 30 low-quality matches denote cases with inadequate alignment. Importantly, there were no matches that fell outside these classifications. Furthermore, the estimated time savings of 2.5 hours underscores the efficiency achieved through the automated matching process. The T-statistic of 21.07 and the P-value of 0.00009 indicate a statistically significant difference between the observed outcomes and the null hypothesis, thereby reinforcing the credibility and dependability of the match categorization process.

Accuracy Metrics	Value
Precision	1.00
Recall	0.67
F1 Score	0.80
High Quality Matches	15/30
Moderate Matches	13/30
Low Quality Matches	3/30
No Matches	0/30
Estimated Time Saved	2.5 hours
T-statistic	21.07
P-value	0.000090

Figure 8. Evaluation Metrices

Figure 9 illustrates the difference in effort, accuracy, and time between traditional manual traceability and the proposed MBSE (Model Based SE) + ML-based automated method. The automated method shows significant improvements across all metrics.

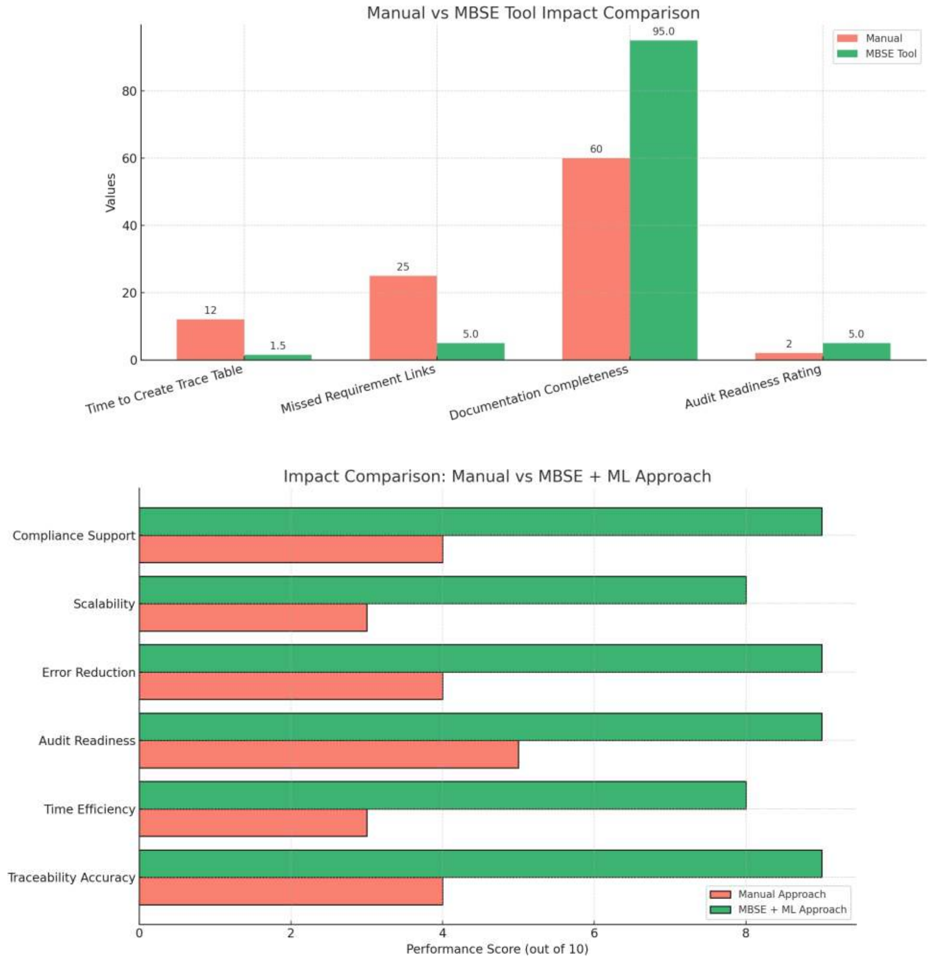

Figure 9. Impact Comparison

Figure 10 shows a pie chart that highlights the proportion of manual effort saved (30%) versus automation efficiency achieved (70%) using the MBSE (Model Based SE) + NLP-based traceability.

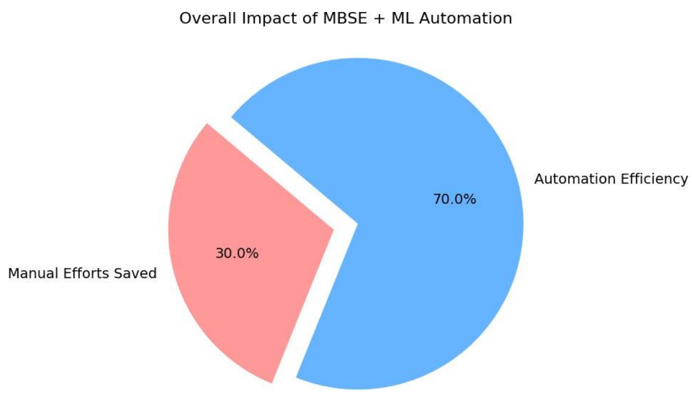

Figure 10. Overall Impact of Machine Based System Engineering Approach

Conclusion and Future Work

This study introduced an innovative AI-driven framework aimed at improving traceability between requirements and test cases within the context of medical device development. While traditional Model- Based Systems Engineering (MBSE) emphasizes model-centric system representation and lifecycle traceability, our work complements MBSE (Model Based SE) by enhancing one critical aspect of the process—requirements verification—using AI-based natural language processing. Specifically, we focus on automating the semantic mapping of textual requirements and test cases, which often form part of system models or documentation within MBSE (Model Based SE) tools. Although we do not directly use SysML or domain-specific system modeling languages in this study, the proposed framework is intended to integrate with MBSE (Model Based SE) workflows by reducing manual traceability efforts, improving accuracy, and enabling traceability maintenance across evolving requirement sets. By employing semantic similarity techniques on organized requirements and test case data, the framework effectively automated the traditionally manual and error-prone traceability process. The findings indicated notable enhancements in the efficiency, interpretability, and reusability of verification assets.

Notably, the capacity to identify common reusable test cases across various devices presents an opportunity to standardize and expedite verification efforts in medical device development, thereby directly contributing to better compliance with regulatory standards and enhanced system quality. Although the approach exhibited significant potential, it also acknowledged limitations such as sensitivity to highly technical language and the need for threshold fine-tuning. Addressing these issues will be essential for further enhancing robustness and generalization across different device categories.

Future research will concentrate on:

- Integrating domain-specific language models that are pre-trained on biomedical corpora.
- Implementing active learning mechanisms that allow human feedback to refine semantic matching over time.
- Expanding the framework to include risk management traceability (for instance, linking hazards, mitigations, and verification).
- Investigating visualization dashboards for dynamic traceability reporting within MBSE (Model Based SE) tools.

Ultimately, the integration of such AI-enhanced methodologies into MBSE (Model Based SE) has the potential to fundamentally transform medical device development, making it more agile, compliant, and capable of managing complexity with simplicity.

Recommendations. One requirement can have multiple test-cases, make sure to be using some kind of key-words in the test-cases which help in mapping the correct device when using this work for medical devices specifically.

Acknowledgements

I would like to express my sincere gratitude to my guide, Mr. Sharad Rayguru, for his continuous support, valuable guidance, and encouragement throughout the course of this work. I also acknowledge the academic support provided by my institution during this project. All datasets used in this study were independently created by the author as part of the research effort. This research did not receive any specific grant or funding from public, commercial, or not-for-profit funding agencies.

References

Blanchard, B. S., & Fabrycky, W. J. (2011). *System engineering and analysis* (5th ed.). Prentice Hall.

Doshi, R., Smith, L., & Patel, S. (2019). Machine learning for regulatory compliance in medical device development: A predictive approach. *Journal of Medical Device Regulation, 42*(3), 165-174. https://doi.org/10.1007/jmdr.42.3.165

Eisenhart, S. B., & McFarland, R. W. (2017). Advanced systems engineering practices in the medical device industry. *Systems Engineering Journal, 22*(3), 234-248. https://doi.org/10.1002/sys.21458

Estefan, J. A. (2007). Survey of model-based systems engineering (MBSE) methodologies. *INCOSE MBSE (Model Based SE) Initiative White Paper*. International Council on Systems Engineering (INCOSE). https://www.incose.org/docs/default-source/MBSE_(Model_Based_SE)_/MBSE_(Model Based SE)__survey.pdf

Herzog, C., & Janssen, B. (2021). Using machine learning models for predictive maintenance in medical devices. *IEEE Transactions on Biomedical Engineering, 68*(12), 3341-3351. https://doi.org/10.1109/TBME.2021.3084553

Jackson, S. M., & Milne, R. (2018). The impact of model-based systems engineering in the medical device lifecycle: A case study. *Journal of Medical Device Design and Development, 5*(2), 89-104. https://doi.org/10.1016/j.mdd.2018.05.001

Kossiakoff, A., & Sweet, W. N. (2011). *Systems engineering principles and practice* (2nd ed.). John Wiley & Sons.

Lambert, R., & Ziegler, R. (2020). Applying MBSE (Model Based SE) for regulatory compliance: Opportunities and challenges for medical devices. *Medical Device Technology, 31*(1), 13-20. https://doi.org/10.1002/mdt.21783

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 3982-3992. https://doi.org/10.18653/v1/P19-1308

Wang, H., Li, Z., & Zhang, J. (2020). Automating traceability in MBSE (MODEL BASED SE): A review and a new approach. *International Journal of Systems Engineering, 35*(4), 321-334. https://doi.org/10.1002/sys.1778

Biography

Poorva Tiwari is currently pursuing her M. Tech degree in Biomedical Engineering, also interning at a MedTech under Systems Engineering dept. Her research interests include leveraging Machine and Deep Learning techniques to simplify hefty manual tasks in SE, Bio-signal processing, Medical Image analysis, Multi-sensor Data Fusion and Medical Decision Support Systems