

Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation

Michael Olusanya ^{1,✉}, Oluwasefunmi Arogundade ¹, Dada O. Aborisade ¹, Musiliu Raji ², Oluwafolake Ojo ³

1. Computer Science, Federal University of Agriculture, Abeokuta, Abeokuta, NGA

2. Mathematics, Federal University of Agriculture, Abeokuta, Abeokuta, NGA

3. Computer Science, Osun State University, Osogbo, Osogbo, NGA

Received: June 17, 2025 | Review began: November 01, 2025 | Review ended: April 14, 2026 | Published: April 22, 2026

© Copyright 2026

This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Software requirements specification (SRS) interpretation is a crucial aspect of software engineering, as it influences the accuracy of requirement analysis, validation, and verification. Traditional topic modeling techniques such as Latent Dirichlet Allocation (LDA) and Correlated Topic Modeling (CTM) often struggle with semantic coherence due to their reliance on probabilistic word distributions without considering contextual relationships. This study introduces Hierarchical Clustering-based Latent Dirichlet Allocation (HCI_LDA), an improved LDA-based model incorporating hierarchical clustering-based initialization, to enhance semantic coherence. The study evaluates HCI_LDA alongside five other SRS interpreters, including CTM, ChatGPT, and human experts, across three SRS datasets of varying sizes: SRS4.0 (2305 words), EXINFM (978 words), and NASA (492 words). Experimental results show that document size influences cosine similarity scores, with HCI-LDA achieving its highest score of 0.52 on the largest document (SRS4.0), compared to 0.10 for EXINFM and 0.32 for NASA. Comparative analysis of HCI-LDA against other LDA-based models shows an average mean improvement score (MIS) of 136.67% over CTM across all datasets, indicating higher semantic alignment with the original SRS documents. A comparison of human experts (H1, H2) and ChatGPT shows variation in similarity scores across datasets, with ChatGPT achieving higher scores in NASA (0.55 vs. 0.43 and 0.33) but lower scores in SRS4.0 and EXINFM. Additionally, comparison with traditional LDA shows substantial increases in similarity scores (MIS: 403.33%), reflecting improved representational correspondence, validating the benefits of clustering-based topic modeling in requirement engineering. These findings emphasize the importance of semantic-aware techniques for software requirement analysis, highlighting the potential for AI-assisted SRS interpretation in software validation and verification. The study has implications for empirical software engineering, industry, and practice, as it demonstrates how clustering-based topic modeling can improve semantic alignment in SRS representations and suggests the potential of such approaches for supporting structured requirements analysis.

Categories: Requirements Engineering, Software Development, Natural Language Processing (NLP)

Keywords: software requirements specification, topic modelling, hierarchical clustering, latent dirichlet allocation, correlated topic modelling, chatgpt

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

Introduction

Software requirement specifications (SRS) serve as the foundational blueprint for developing a software product, making them critically important to developers and the overall success of the software. At the heart of this significance lies the clarity and precision they provide, acting as a bridge between stakeholders' expectations and the technical implementation. By delineating the functional and non-functional requirements (NFRs), SRS guides developers in understanding what the software should accomplish and how it should perform. This initial clarity is crucial for making informed decisions about project scope, resource allocation, and timeline estimation. Moreover, SRS serves as a baseline for assessing project feasibility and aligning stakeholders' expectations with technical capabilities [1]. As the development phase commences, SRS becomes the reference point for developers to translate requirements into design specifications and code. Unambiguous SRS documentation reduces ambiguity and minimizes the risk of misinterpretation or divergent implementations. It ensures that developers stay focused on delivering features that align with business goals and user expectations [2]. Furthermore, SRS plays a pivotal role in the quality assurance and testing phases. Test cases and scenarios are derived directly from the SRS, facilitating comprehensive validation of software functionality against specified requirements. Any deviations or discrepancies identified during testing can be traced back to the SRS, enabling efficient debugging and corrective action [3]. Beyond the development cycle, SRS continues to be relevant during maintenance and evolution phases. It serves as a reference for understanding the rationale behind design decisions, making enhancements, and addressing change requests. A well-documented SRS streamlines ongoing support and maintenance activities by providing a clear roadmap of the software's intended behavior and capabilities.

In the requirement engineering industry, the process of producing the SRS document typically involves collaboration among various stakeholders, including business analysts, project managers, domain experts, and end users [4]. However, the responsibility of coining the words and articulating the requirements primarily rests with business analysts and domain experts who possess a deep understanding of both the business domain and technical aspects of the software [5]. Developers must thoroughly comprehend these requirements to ensure accurate implementation and alignment with stakeholders' expectations. To achieve a thorough understanding of functional requirements and user expectations as described in the specification document, software developers employ several approaches. Therefore, interpreting SRS, particularly the functional requirements, is a critical aspect of software development for developers [6].

Besides human efforts, having a dedicated go-to tool for in-depth understanding of SRS would streamline the interpretation process, foster consistency, and enhance efficiency in translating functional requirements from the SRS into technical implementations. Moreover, artificial intelligence (AI) approaches are divers in literature. AI readily plays significant roles in the software development process, particularly in aiding developers and teams in various tasks [7]. In the context of SRS interpretation, natural language processing (NLP) can be utilized to perform sentiment analysis [8], extract entities and relationships [9], classify requirements based on priority or category, and generate summaries or abstracts of lengthy SRS documents [8]. Within NLP, topic modeling algorithms are particularly relevant for identifying and extracting topics or themes from SRS documents. Topic modeling algorithms such as latent Dirichlet allocation (LDA) can automatically discover patterns and group-related requirements, aiding developers in understanding the overarching themes and functional areas covered in the SRS [10].

Effective topic modeling is essential for extracting meaningful insights from large text datasets, yet traditional LDA suffers from several limitations, including random initialization of topic-word distributions, which often leads to poor topic coherence and instability in results [11]. Many existing approaches fail to capture contextual word relationships, leading to topics that are difficult to interpret or irrelevant to the underlying text [12]. Additionally, existing semantic-based improvements to LDA have largely focused on incorporating external ontologies or pre-defined semantic relationships, which may not accurately reflect the context-specific relationships within a given dataset and also encounter challenges in handling ambiguous words and ensuring that discovered topics remain contextually relevant. Without an effective initialization strategy, LDA struggles to consistently generate meaningful, well-separated, and semantically coherent topics [13].

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

In this study, a clustering-based initialization to refine topic modeling for SRS interpretation was introduced. By addressing the limitations of traditional LDA, this study seeks to develop a more context-aware and semantically coherent approach, ultimately improving software engineers' ability to extract meaningful insights from textual data. In addition, evaluation of the performance of other techniques in distilling and interpreting SRS will be conducted by employing similarity index computations that accurately measure their semantic alignment and contextual relevance to the actual content of the SRS documents.

Literature review

Dakshi et al. [11] incorporated an ontology-driven approach called Semantic-LDA to map words to concepts dynamically, rather than relying on static ontology values. The model quantified concept-word relationships using word distributions from the input dataset, ensuring that the generated topics were more aligned with the document context. Furthermore, an enhanced version of Semantic-LDA was developed to handle ambiguous words, allowing the model to better interpret words without direct ontology matches. To evaluate the effectiveness of the proposed model, the study conducted experiments comparing Semantic-LDA and its enhanced version against state-of-the-art topic modeling systems. The evaluation focused on two key performance metrics: topic quality and information filtering. The results demonstrated that Semantic-LDA significantly outperformed baseline models, confirming that the integration of semantic understanding improved the coherence, relevance, and interpretability of discovered topics [11].

Saklamaeva and Pavlič [14] investigate the role of AI-driven assistants within the context of scaled agile development methods (SADMs), particularly within the widely adopted Scaled Agile Framework (SAFe). The paper has three main objectives: evaluating the challenges and impediments organizations face when implementing SADMs, assessing the benefits of incorporating AI in large-scale contexts, and identifying aspects of SADMs that AI-driven assistants enhance. Through a comprehensive systematic literature review, the authors identified 18 distinct challenges that organizations encounter. They also pinpointed seven benefits and five challenges related to AI implementation in SADMs. Additionally, the paper examines 15 different AI-driven assistants and tools, analyzing their effectiveness in addressing the identified challenges. The findings highlight the exceptional versatility and effectiveness of AI-driven assistants, demonstrating their capability to tackle a broad range of problems [14].

Rieger et al. [13] proposed LDAPrototype, a method designed to improve the reliability of topic modeling by selecting the most representative LDA output from several repeated runs on the same dataset. Unlike traditional LDA selection approaches that rely on metrics like perplexity or NPMI, LDAPrototype uses a custom similarity metric known as S-CLOP (Similarity of multiple sets by Clustering with Local Pruning) to evaluate topic alignment across replications. This strategy helps ensure more consistent and reproducible topic structures. However, one noted drawback is the computational intensity of the repeated modeling process, especially when applied to large datasets or models with many topics [13].

Mohamed et al. [15] focus on the detection and correction of pragmatic ambiguity in SRS. Pragmatic ambiguity arises when a statement can have more than one interpretation, leading to potential misinterpretations and incorrect software implementations. The study employs word embedding techniques to identify ambiguous terms and uses linked data to resolve these ambiguities. The aim is to estimate the degree of ambiguity for common computer science terms when used across different domains. To validate their tool, the authors used open-source SRS documents and compared the tool's performance against human capabilities. The results indicated that the tool was more accurate in detecting and correcting pragmatic ambiguities than humans, especially when dealing with terms across different domains [15].

Darnoto and Siahaan [16] introduce MultiPhiLDA, a modified version of LDA, designed specifically for detecting irrelevant statements in SRS. MultiPhiLDA addressed this challenge by enhancing LDA to identify actors and actions within SRS, thereby improving the detection of irrelevant requirement statements. The expected outcome is an automated system capable of assisting analysts in identifying and filtering out irrelevant statements from SRS, ultimately contributing to the development of high-quality software products [16].

The study by Oran et al. [17] introduces the ReComP framework, designed to evaluate and improve the communication of requirements specifications from the perspectives of developers and testers. Recognizing that different roles within a software development team have varying informational needs, ReComP aims to ensure that requirements specifications

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

provide the necessary information to reduce errors stemming from inadequate communication. This study targets intra-team communication. Using the Design Science Research method, the authors developed and refined ReComP through two DSR cycles, focusing on user stories for developers and use cases for testers. The study focuses on user stories and use cases, which might limit its applicability to other forms of requirements specifications used in different methodologies or project types [17].

The study by Mucahit et al. [18] explores the integration of NLP techniques with SRS to enhance the software development process. The research focuses on leveraging NLP for requirement classification, document categorization, and topic modeling within SRS documents. The study also highlights the potential of NLP to delve into the deeper semantics of software requirements, aiding in tasks such as information extraction, automation of UML diagrams, AI-enabled requirement analysis, and the detection of defects and ambiguities in SRS. The reliance on tools like IBM DOORS may limit the generalizability of the findings, as different organizations may use varying requirement management tools, necessitating tailored NLP solutions [18].

Younas et al. [19] present an approach for the extraction of NFRs from software requirement documents, leveraging semantic similarity distance based on co-occurrence patterns in large human knowledge repositories like Wikipedia. The study acknowledges the equal importance of both functional and NFRs in software development, noting the challenge of extracting NFRs when they are intermixed with functional requirements in natural language expressions. This approach aims to overcome the drawbacks associated with supervised learning, such as domain-specific training and retraining requirements. The proposed approach utilizes semantic similarity calculations to identify NFRs by analyzing the textual semantics of functional requirements and comparing them with popular indicator keywords associated with NFRs. Additionally, the research explores three pre-processing approaches (traditional, part-of-speech tagging, and word augmentation) to enhance NFR extraction performance [19].

Ekinci et al. [20] proposed a semantic network based LDA model. A semantic measure is obtained from several semantically similar documents. This measure is introduced into the proposed model to influence document-topic distribution. To obtain semantically similar documents, the feature space of documents is constructed by concepts and named entities instead of the words that compose the documents. The representation of the feature space with these units ensures that the semantic knowledge is correctly included in the model. The experimental results of the NET-LDA model indicate that injection of semantic similarity to the model provides semantically related, coherent, and meaningful topics [20].

Priyadi et al. [21] address the importance of understanding interdependencies among requirements in SRS. Recognizing that SRS documents contain both explicit and implicit information about these dependencies, the authors propose an approach to model a requirement dependency graph. This graph captures relationships such as similarity, part-of, and elaboration among requirements. The methodology involves text preprocessing steps, including tokenization, stopword removal, and stemming. Semantic similarity is measured using WS4J (WordNet Similarity for Java) and methods like Wu Palmer and Levenshtein. The extraction results are then optimized using greedy algorithms. Finally, the reliability of the generated dependency graph is assessed using Gwet's AC1 formula [21].

Prior studies demonstrate substantial progress in applying topic modeling, semantic enrichment, and NLP techniques to SRS; a significant proportion of existing approaches extend traditional LDA-based models. However, LDA is inherently sensitive to random initialization and hyperparameter configurations, often resulting in unstable topic distributions across different runs. While some studies attempt to improve coherence or reproducibility, few directly address the fundamental instability introduced at the initialization stage, particularly in domain-specific corpora such as SRS.

In addition, much of the literature focuses on isolated SRS analysis tasks, including ambiguity detection, filtering irrelevant requirements, extracting NFRs, modeling requirement dependencies, and communication enhancement frameworks. Although these contributions are valuable, they are largely task-specific and operate in isolation. Consequently, they do not provide a unified semantic representation that supports comprehensive SRS interpretation.

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

Relatively limited attention has been given to constructing a robust semantic structure that simultaneously enhances topic stability, captures hierarchical relationships inherent in SRS documents, and provides a unified semantic representation that can support different SRS interpretation tasks, rather than addressing them in isolation.

Materials And Methods

The conceptual framework of this study is presented in Figure 1. The framework comprises four broad phases, including the SRS corpus acquisition, interpretations, vector extraction through Word2Vec, and the cosine similarity index computations. In the second phase, interpretations are further divided into the human interpretation phase, the topic modeling phase, and the generative AI phase. The expertise of two human domain experts is sought, and two topic modeling models of LDA and Correlated Topic Model (CTM) are deployed. This study likewise introduces a modified version of the LDA, the Hierarchical Clustering (HCI)-based topic modeling, to make three in that category. Therefore, ChatGPT (generative AI) makes the six interpreter whose interpretations would be evaluated and compared with the other five to investigate the most suitable technique for SRS modeling.

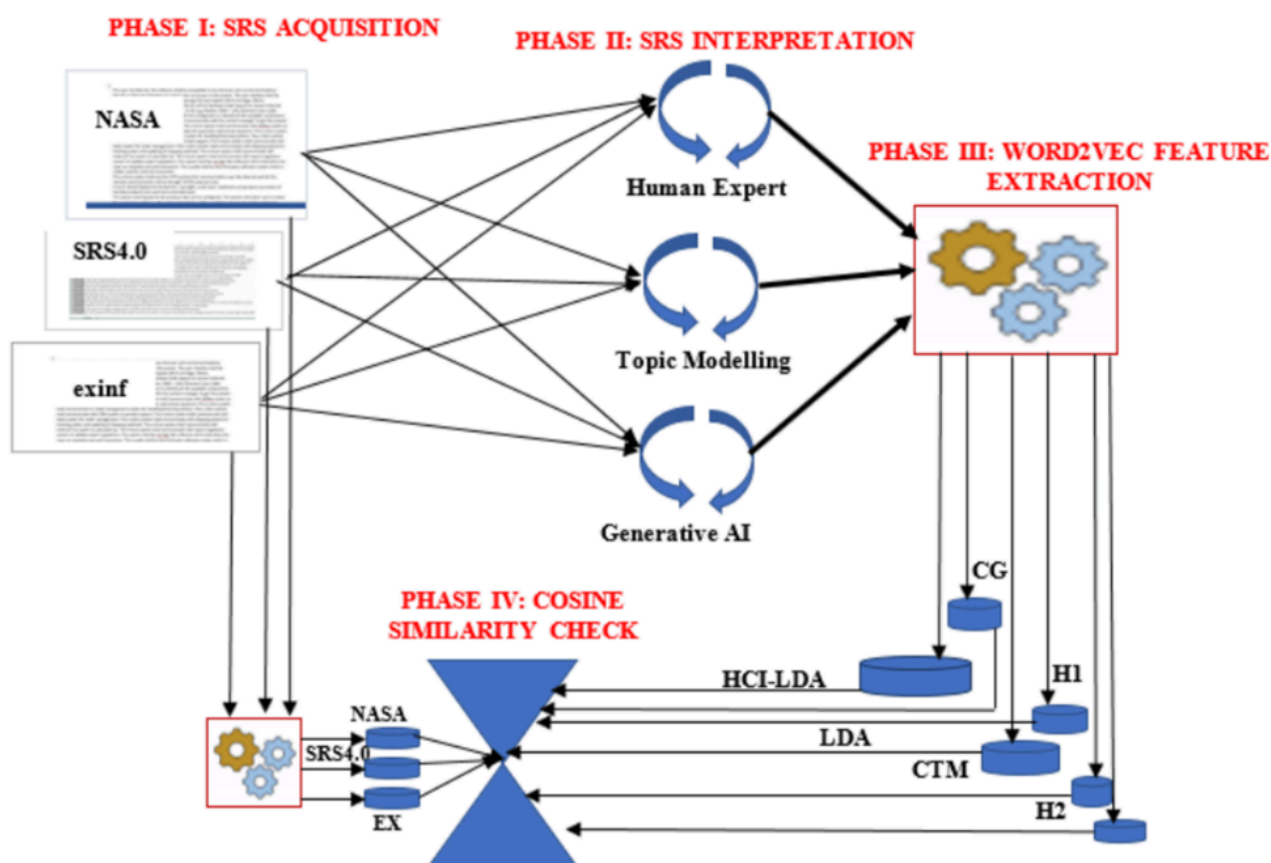


FIGURE 1: The conceptual framework of the proposed methodological approach

CG, ChatGPT; CTM, Correlated Topic Modeling; H1, Human Expert 1; H2, Human Expert 2; HCI, Hierarchical Clustering; LDA, Latent Dirichlet Allocation; SRS, Software Requirements Specification

Phase 1 - SRS acquisition

SRS acquisition is the first phase of the framework. The documents employed in this study were downloaded from three digital libraries comprising functional and non-functional requirements of software products. The acquired data are from software engineering-based repositories, including NASA (NASA SRS), SRS4.0 (SRS4.0), and exinfmSRS (EXINFMSRS).

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

The varying characteristics of the acquired SRS court across business management, aerospace and educational use cases, as well as varying corpus sizes, including large (2,000+), medium size (neighborhood of 1,000) and small size in the neighborhood of 500 words. The acquired documents are analyzed by the proposed techniques in the subsequent phase.

Phase 2 - Interpretations

This phase describes the process of the SRS distillation by the techniques. Each of the techniques would be treated with the three data categories presented.

The first technique is the human professionals who are software developers. Two human evaluators are deployed for this study. They are domain experts with diverse experiences in software development and are familiar with the task of interpreting SRS before software development. They are required to take in the three datasets and present their interpretations as regards their understanding of the end user specifications within the SRS document. They are expected to bring in their wealth of experience by returning germane understanding of their insights into the provided SRSs.

The second technique for SRS distillation for this study is ChatGPT, based on the Generative Pre-trained Transformer architecture. It is a language model designed by OpenAI. Its methodology revolves around transformer-based models, specifically designed for natural language understanding and generation tasks. Using ChatGPT for interpreting SRS involves providing prompts or inputs related to the SRS content and allowing ChatGPT to generate relevant topics or summaries based on the provided information. Based on the processed input, ChatGPT generates text that is relevant to the prompts, potentially including topic summaries, insights, or explanations related to the SRS content. The model's generative capabilities enable it to produce coherent and contextually relevant responses based on the input it receives.

The third technique for SRS distillation for this study is topic modeling algorithms. This study introduces HCI-LDA. The performance of HCI-LDA is compared with two traditional topic modeling algorithms, LDA and CTM. The three would make the techniques from the Topic Modeling family. The LDA is a generative probabilistic model for topic modeling, which assigns topics to documents based on word distributions, while the CTM is unique in its approach as it incorporates correlations between topics, and captures dependencies between topics in documents.

Derived Hierarchical Clustering-Based Topic Modeling

For this study, a data-driven initialization of LDA is presented using hierarchical clustering to present a variant of the LDA model (HCI_LDA) whose initialization is structured. The derived model, HCI_LDA, is expected to enhance traditional LDA for the software requirement specifications summarization task by replacing the conventional random topic initialisation of LDA with a structured, data-driven approach. Instead of randomly assigning words to topics, HCI_LDA, as shown in Figure 2, applies hierarchical clustering on word embeddings to group semantically similar words. These clusters serve as initial topic assignments, ensuring that words with related meanings are likely to be grouped from the start. This initialization would improve topic coherence and convergence speed, leading to more interpretable topic distributions for summarizing SRS documents effectively. The random initialization approach in traditional LDA, which is based on probabilistic modeling, is then replaced with a structured hierarchical clustering approach in this study, named the HCI_LDA model.

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

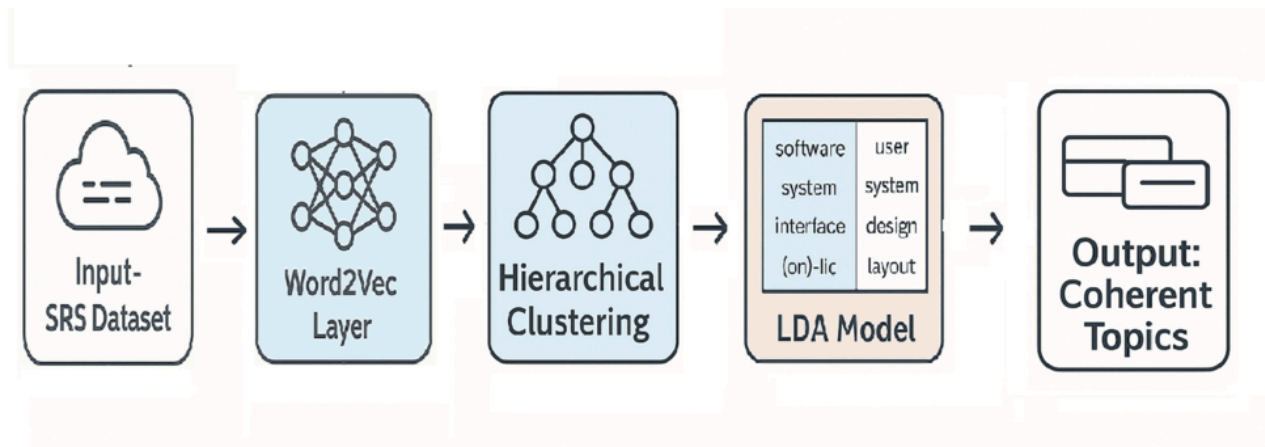


FIGURE 2: The architecture of the derived HCI-LDA

SRS, Software Requirements Specification; HCI-LDA, Hierarchical Clustering-based Latent Dirichlet Allocation

Phases of the HCI_LDA Model

1. Feature Extraction Using Word2Vec: The input, SRS datasets, undergoes preprocessing steps such as tokenization, lowercasing, stop word removal, and lemmatization to ensure semantic clarity. The cleaned text is then transformed into dense vector representations using the Word2Vec model, effectively capturing contextual and semantic relationships. These embeddings form the semantic basis for HCI-LDA.

Let $D = \{d_1, d_2, \dots, d_N\}$ be the corpus of SRS, where each document d_i consists of a sequence of words $w_{i1}, w_{i2}, \dots, w_{iM}$

Word embeddings: Convert each word w_j into a dense vector using Word2Vec:

$$V_j = \text{Word2Vec}(w_j) \quad (1)$$

where $v_j \in \mathbb{R}^d$ is a d -dimensional word vector

Document representation: Compute the document vector v_{d_i} as the mean of word vectors:

$$v_{d_i} = \frac{1}{M} \sum_{j=1}^M v_j \quad (2)$$

where M is the number of words in document d_i .

2. Hierarchical Clustering for Initialization: To improve the quality and stability of topic model initialization, a hierarchical clustering approach is applied to the semantic word embeddings generated from the SRS corpus. A pairwise distance matrix is first constructed using cosine similarity to quantify the semantic proximity between word vectors. Subsequently, agglomerative hierarchical clustering is performed using Ward's linkage method, which iteratively merges word clusters in a manner that minimizes the total within-cluster variance. This clustering process produces semantically coherent groups of terms that serve as informed topic priors, effectively replacing the random initialization typically used in conventional LDA models. By embedding semantic-aware structure into the initialization phase, this step lays a more meaningful foundation for accurate and interpretable topic discovery.

Instead of random topic initialization, hierarchical clustering groups similar documents.

Pairwise distance matrix: Compute the cosine similarity between document vectors:

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

$$S(d_i, d_j) = \frac{\mathbf{v}_{d_i} \cdot \mathbf{v}_{d_j}}{\|\mathbf{v}_{d_i}\| \|\mathbf{v}_{d_j}\|} \quad (3)$$

Agglomerative clustering: Perform hierarchical clustering with Ward's linkage:

$$d(i, j) = \|\mathbf{v}_{d_i} - \mathbf{v}_{d_j}\|^2 \quad (4)$$

where clusters are merged based on minimizing intra-cluster variance.

Topic initialization: Assign each initial cluster as a topic, and initialize topic-word distributions.

3. LDA With Clustering-Based Initialization: The LDA generative model is employed, with topic-word distributions initialized based on the output of hierarchical clustering rather than random assignment. This structured initialization improves the convergence and stability of the model. Topic inference and parameter estimation are conducted using Gibbs sampling, enabling efficient posterior approximation.

LDA models documents as a mixture of topics, where each topic is a distribution over words.

Generative process of LDA: Each document d_i is associated with a topic distribution θ_i where:

$$\theta_i \sim \text{Dirichlet}(\alpha)$$

Each topic z_k is a distribution over words:

$$\theta_k \sim \text{Dirichlet}(\beta)$$

For each word in document d_i :

Choose a topic $z_{i,j} \sim \text{Multinomial}(\theta_i)$

Generate word $w_{i,j}$ from the topic-word distribution:

$$w_{i,j} \sim \text{Multinomial}(\phi_{z_{i,j}})$$

Parameter estimation using Gibbs sampling: Compute topic assignment probabilities:

$$P(Z_{(i,j)} = k \mid w_{(i,j)}, \theta, \phi) \propto P(w_{(i,j)} \mid \phi_k) P(z_{(i,j)} \mid \theta_i) \quad (5)$$

Update θ_i and ϕ_k

$$\theta_i = \frac{n_{i,k} + \alpha}{\sum_w (n_{k,w} + \alpha)} \quad (6)$$

$$\phi_k = \frac{n_{k,w} + \beta}{\sum_w (n_{k,w} + \beta)} \quad (7)$$

where, $n_{i,k}$ is the number of words in document d_i assigned to topic k , and $n_{k,w}$ is the number of times word w is assigned to topic k

4. Through iterative sampling via Gibbs sampling, HCI-LDA incrementally refines topic assignments while preserving the semantic structure introduced during initialization. This process enables the model to uncover latent topics that are both thematically coherent and semantically meaningful.

Final model formulation:

How to cite this article:

$$\text{LDA-HCI}(D, K) = \arg \max \prod_{i=1}^N P(d_i | \theta_i, \emptyset) \quad (8)$$

subject to:

$$\theta_i \sim \text{Dirichlet}(\alpha), \quad \phi_k \sim \text{Dirichlet}(\beta)$$

with hierarchical clustering prior topic assignments.

Algorithm: Data-Driven Hierarchical Clustering-Based Initialization for LDA

Input:

$D = \{d_1, d_2, \dots, d_N\} \rightarrow$ Set of SRS documents

$K \rightarrow$ Number of topics

$d \rightarrow$ Dimension of Word2Vec embeddings

$\alpha, \beta \rightarrow$ Dirichlet hyperparameters for LDA

Output:

$\Theta \rightarrow$ Initial document-topic distribution

$\Phi \rightarrow$ Initial topic-word distribution

1. Train Word2Vec on SRS corpus to obtain word embeddings
2. For each document $d_i \in D$, retrieve word vectors
3. Compute document vector v_{d_i} as the average of all word vectors
4. Compute pairwise cosine similarity between document vectors
5. Convert similarity into distance matrix
6. Perform hierarchical clustering
7. Extract K clusters from the hierarchical clustering dendrogram
8. Assign each document in cluster C_k to topic z_k
9. Compute initial topic-word distributions:

$$n_{k,w} = \sum_{d_i \in C_k} \sum_{w_j \in d_i} 1(w_j = w) \quad (9)$$

10. Normalize topic-word probabilities:

$$\phi_k(w) = \frac{n_{k,w} + \beta}{\sum_w (n_{k,w} + \beta)} \quad (10)$$

11. Compute document-topic distributions:

$$\theta_i(k) = \frac{n_{i,k} + \alpha}{\sum_n (n_{i,k} + \alpha)} \quad (11)$$

12. Initialize LDA model with ϕ_k and θ_i
13. Run Gibbs sampling or variational inference to refine topics
14. Output ϕ and θ as final model parameters

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

The algorithm begins by training Word2Vec on the SRS corpus to obtain semantic word embeddings. Document vectors are created by averaging word embeddings, and hierarchical clustering is applied to group documents into K clusters, which serve as initial topic assignments. Topic-word distributions are estimated from these clusters and used to initialize the LDA model. Finally, Gibbs sampling or variational inference refines the topics, resulting in improved topic coherence and semantic alignment in the final document-topic and topic-word distributions.

Phase 3 - Word2Vec feature extraction

The interpretation results produced by the six interpreters, HA1, HA2, LDA, CTM, derived HCI_LDA, and CG, are transformed into word embeddings using a pre-trained Word2Vec model. Through transfer learning, the trained model is activated to obtain vector representations for each interpreter's output. These generated embeddings are then compared with those of the original three SRS datasets to assess semantic alignment. To quantify the degree of alignment, cosine similarity is computed between the respective vector representations.

Phase 4 - Similarity index computations

The cosine similarity index is computed between the pair of HA1 interpretation and the actual SRS, HA2 interpretation and the actual SRS, LDA interpretation and the actual SRS, CTM interpretation and the actual SRS, and the CG interpretation and the actual SRS. The similarity index is calculated using the following formula:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (12)$$

where, $A \cdot B$ represents the dot product of vectors A and B (the pair of each technique and SRS), and $\|A\|$ and $\|B\|$ represent the Euclidean norms of vectors A and B (the pair of each technique and SRS), respectively.

This step measures the degree of similarity between the outputs of different techniques and the original SRS based on their representations. Cosine similarity is used to quantify lexical-semantic alignment between the generated representations and the original SRS, reflecting textual correspondence. A heatmap visualization is generated using `seaborn.heatmap()` to present the cosine similarity values.

Relevance of each process

- i. Before computing cosine similarity, each technique's interpretation or topic needs to be converted into a vector representation using Word2Vec. Word2Vec creates vector embeddings for words based on their semantic meanings in a given context.
- ii. After obtaining vector representations for the technique's interpretations/topics and the SRS document, the cosine similarity calculation quantifies how closely the interpretations align with the actual content of the SRS document.

Results

Evaluation of techniques on SRS documents

The interpretation results returned by the six interpreters were compared with the original three datasets to assess semantic alignment in the embedding space. The cosine computation is implemented as discussed in the methodology of this study to achieve that, and their results are presented in a heat map for interpretation.

The heat map in Figure 3 shows the similarity index of the six interpreters with respect to the SRS4.0 requirement specification document. As can be observed, the results indicate that HCI-LDA exhibited the highest similarity to SRS4.0 with a cosine similarity score of 0.52, followed by ChatGPT (0.23), Human Expert I (0.26), Human Expert II (0.25), and CTM (0.22). LDA, however, showed no similarity to SRS4.0, with a score of 0.00. Comparing techniques among themselves, HCI-LDA also showed notable alignment with CTM (0.30), while ChatGPT exhibited moderate similarity with Human Expert I (0.18) and Human Expert II (0.34). The highest inter-technique similarity was recorded between the two human experts (0.46), suggesting a strong conceptual alignment in their topic distributions.

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

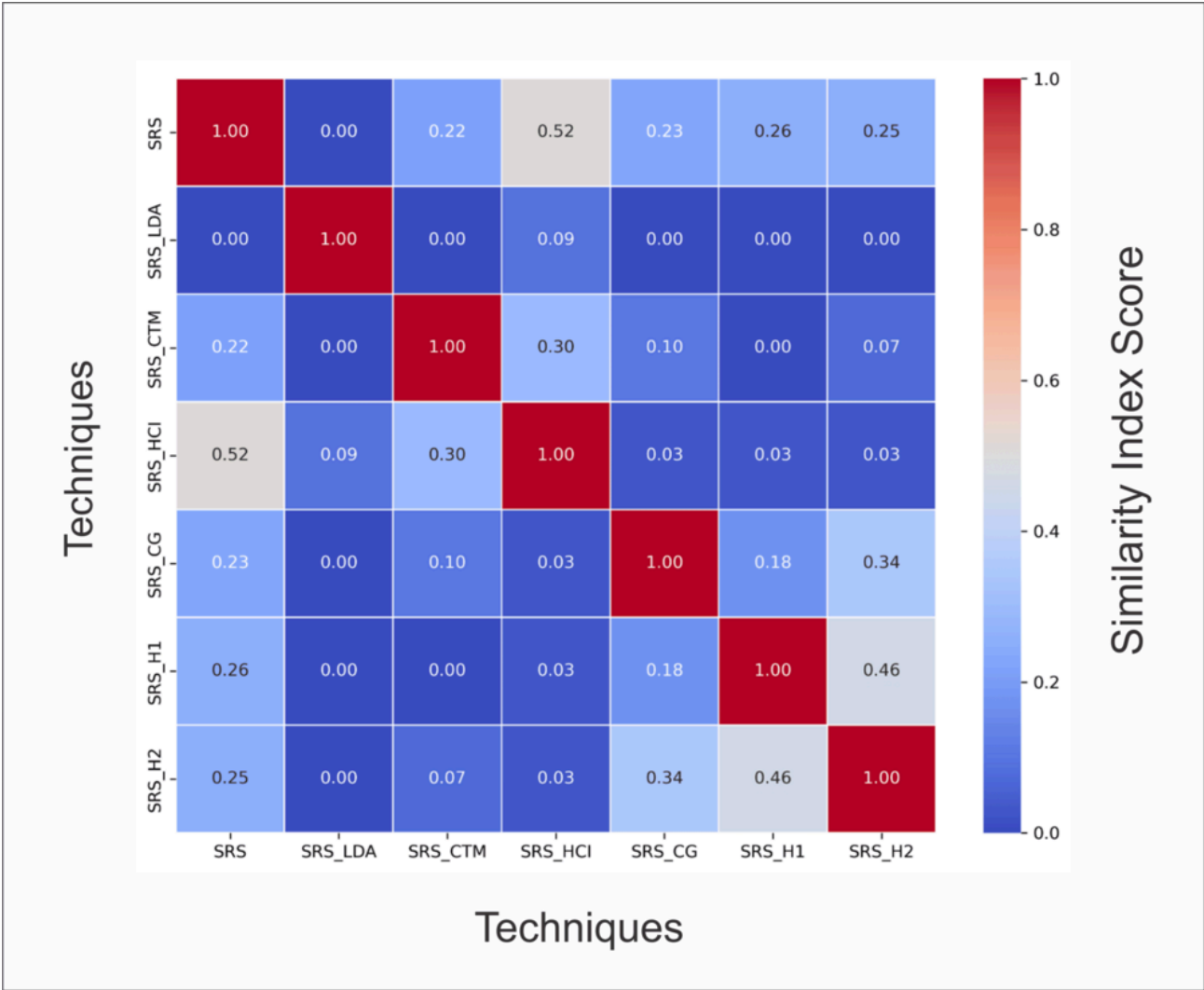


FIGURE 3: Heat map visualization of similarity index between SRS4.0 and six interpreters

CG, ChatGPT; CTM, Correlated Topic Modeling; H1, Human Expert 1; H2, Human Expert 2; HCI, Hierarchical Clustering; LDA, Latent Dirichlet Allocation; SRS, Software Requirements Specification

The cosine similarity between the EXINFMSRS SRS document and the six interpreters was also analyzed to determine their alignment with the original SRS document, and the result of the similarity index is presented in the heat map of Figure 4. The results indicate that ChatGPT exhibited the highest similarity to EXINFMSRS, with a score of 0.42, followed closely by Human Expert I (0.23) and Human Expert II (0.28). HCI-LDA (0.10) and CTM (0.07) displayed lower alignment with EXINFMSRS, while LDA showed no similarity (0.00). Among the interpreters, the highest inter-technique similarity was recorded between Human Expert I and Human Expert II (0.61), followed by ChatGPT and Human Expert II (0.43), and ChatGPT and Human Expert I (0.42).

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

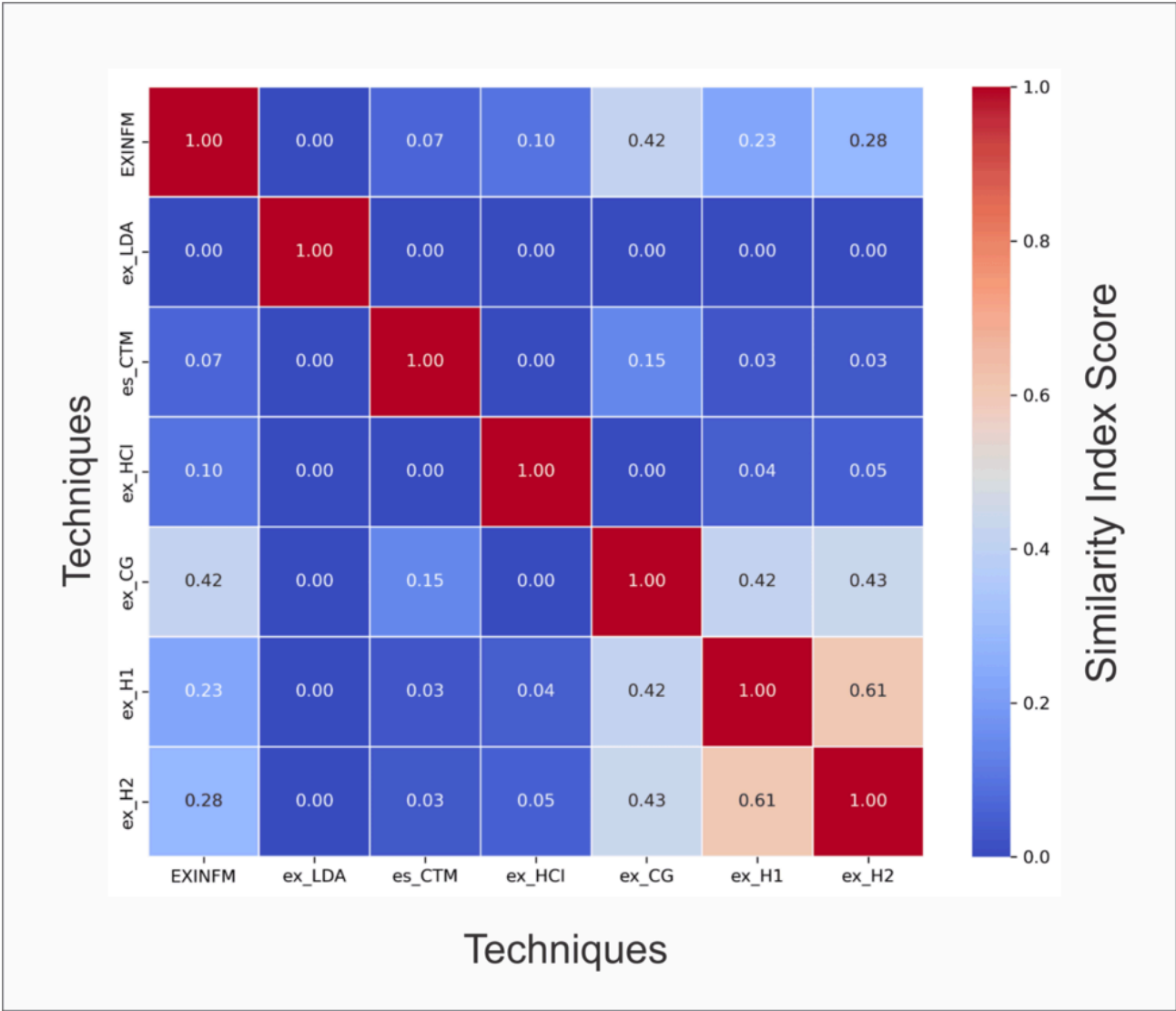


FIGURE 4: Heat map visualization of similarity index between EXINFM SRS and the six interpreters

CG, ChatGPT; CTM, Correlated Topic Modeling; H1, Human Expert 1; H2, Human Expert 2; HCI, Hierarchical Clustering; LDA, Latent Dirichlet Allocation; SRS, Software Requirements Specification

On the NASA dataset, the heat map of Figure 5 shows that ChatGPT demonstrated the highest similarity with NASA SRS (0.55), followed by Human Expert I (0.43) and Human Expert II (0.33). HCI-LDA also showed a moderate alignment (0.32), while CTM (0.14) and LDA (0.00) exhibited little to no similarity. Among the interpreters, Human Expert I and Human Expert II recorded an inter-expert similarity of 0.33, while ChatGPT showed a significant overlap with Human Expert I (0.30). These results suggest varying degrees of alignment between automated techniques, human expertise, and the NASA SRS content.

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

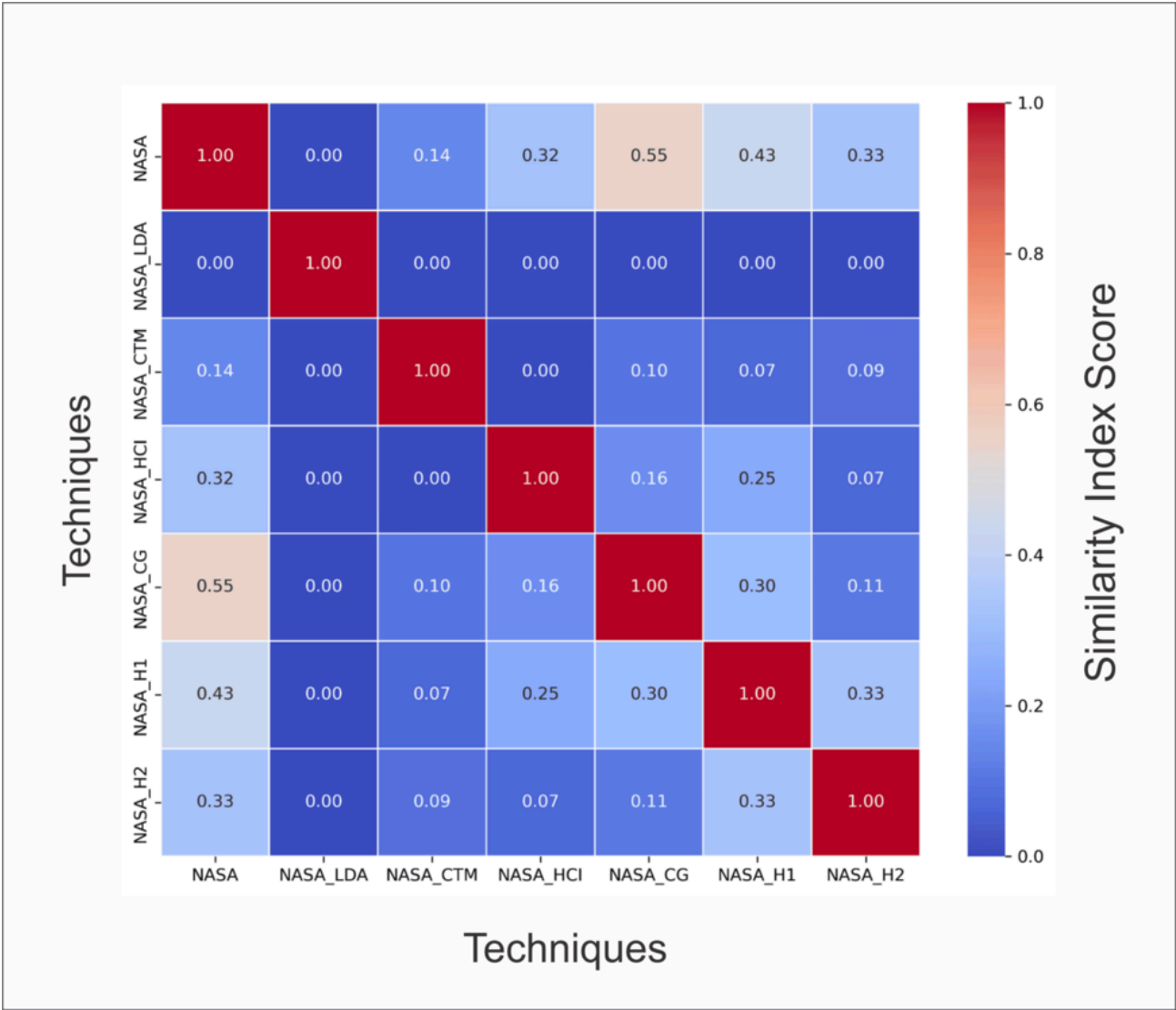


FIGURE 5: Heat map visualization of similarity index between NASA SRS and the six interpreters
CG, ChatGPT; CTM, Correlated Topic Modeling; H1, Human Expert 1; H2, Human Expert 2; HCI, Hierarchical Clustering; LDA, Latent Dirichlet Allocation; SRS, Software Requirements Specification

Comparative thematic interpretation of the experimental result

The experimental result is discussed based on a comparative thematic factor, which could offer insights into the implications of the result.

HCI-Based LDA vs. Traditional Topic Modeling

The results demonstrate that HCI-LDA consistently achieves higher cosine similarity scores than CTM and LDA across all three SRS documents, indicating closer lexical-semantic alignment with the original texts.

Figure 6 shows that HCI-LDA, leveraging hierarchical clustering-based initialization, achieved the highest mean cosine similarity score of 0.31, compared to CTM (0.14) and LDA (0.00), indicating closer lexical-semantic alignment with the original SRS documents. The clustering-based initialization appears to support improved alignment by grouping related words prior to LDA processing. Performance gains of 121.4% over CTM and 3100% over LDA were observed in similarity scores, suggesting stronger representational correspondence across diverse SRS documents, including NASA SRS (492

words), SRS4.0 (2305 words), and EXINFMSRS (978 words). Notably, the higher similarity score for HCI-LDA in SRS4.0 (0.52) indicates that clustering-based initialization may be particularly effective for larger, information-dense documents. Results indicate that better initialization leads to higher cosine similarity scores.

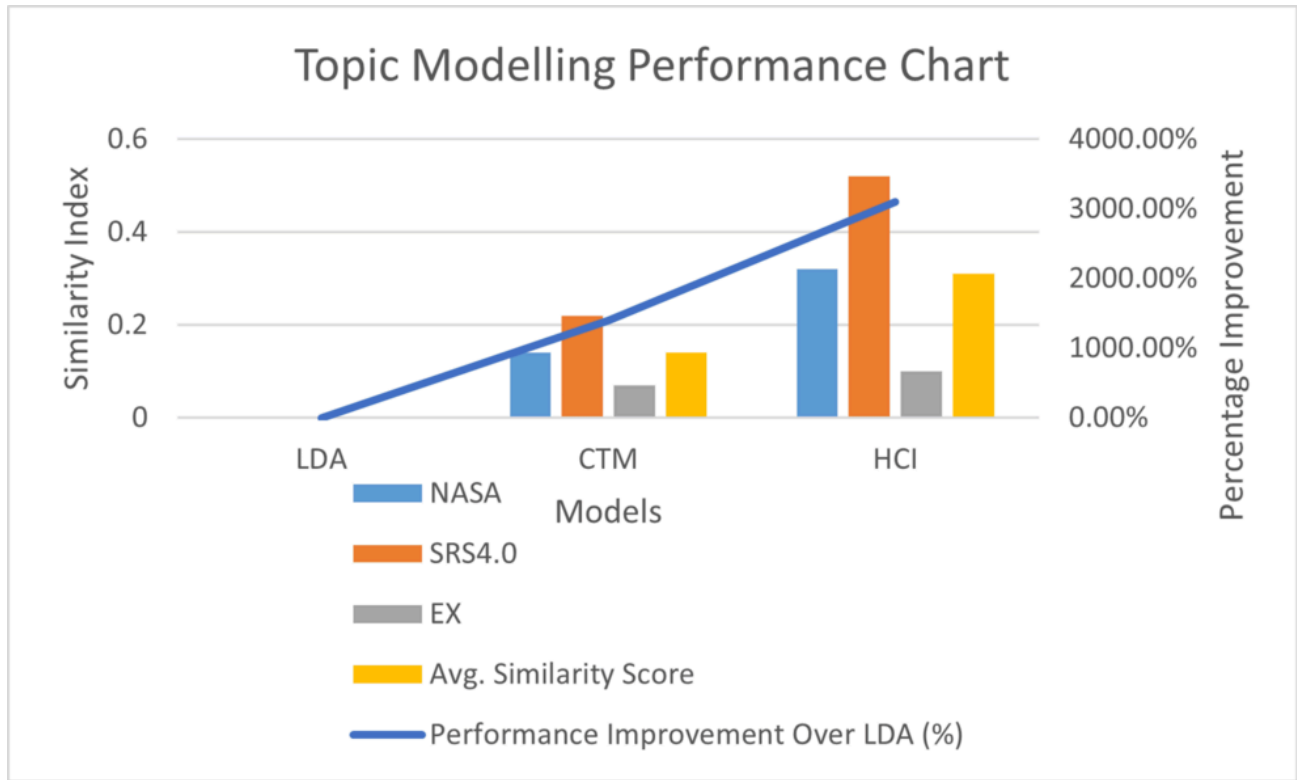


FIGURE 6: Performance chart of the topic modeling-based SRS interpreters

CTM, Correlated Topic Modeling; LDA, Latent Dirichlet Allocation; SRS, Software Requirements Specification; HCI, Hierarchical Clustering

LDA's primary limitation lies in its reliance on random initialization, which can lead to variability in the generated topic representations, particularly for structured technical documents such as SRS. CTM extends LDA by modeling topic correlations; however, it also depends on initialization and may exhibit variability in its outputs.

In contrast, HCI-LDA employs hierarchical clustering to pre-group related words before applying LDA, resulting in more structured initial representations and reduced randomness. This approach is associated with higher cosine similarity scores, indicating closer lexical-semantic alignment with the original SRS documents and more consistent representational outcomes across experiments. Overall, these results suggest that initialization strategies can influence the alignment of topic representations in structured requirements documents.

As can be observed from Table 7, a relative performance (%) and mean improvement over HCI scores are computed from the similarity scores of the interpreters. The performance of each model relative to the document size is computed as:

$$\frac{\text{Model Score}}{\text{Document Size}} \times 100 \tag{13}$$

This is to show how well each model scales with increasing document length. The mean improvement over HCI score represents the percentage change in performance of an interpreter relative to HCI:

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

$$\frac{\text{HCI Score} - \text{Model Score}}{\text{HCI Score}} \times 100$$

(14)

This indicates how much better or worse a model performs compared to the HCI baseline.

Dataset	SRS4.0	EXINFMSRS	NASA	MIH (%) (SRS4.0)	MIH (%) (EX)	MIH (%) (NASA)
Document Size	2305	978	492	-		
LDA	0.00	0.00	0.00	-100.00	-100.00	-100.00
CTM	0.22	0.07	0.14	-57.69	-30.00	-56.25
HCI	0.52	0.10	0.32	-		
ChatGPT	0.23	0.42	0.55	-55.77	+320.00	+71.88
Human Expert I	0.26	0.23	0.43	-50.00	+130.00	+34.38
Human Expert II	0.25	0.28	0.33	-51.92	+180.00	+3.13

TABLE 1: Comparative analysis of SRS document size with respect to the performance of derived HCI

CTM, Correlated Topic Modeling; HCI, Hierarchical Clustering; LDA, Latent Dirichlet Allocation; MIH, Mean Improvement over HCI; SRS, Software Requirements Specification

As observed, ChatGPT achieves higher cosine similarity scores than HCI-LDA, with increases of 320% for EXINFMSRS and 71.88% for NASA SRS, indicating closer lexical-semantic alignment with the original documents in these cases. However, its performance on SRS4.0 is 55.77% lower, suggesting reduced alignment for longer, more complex documents. Human Experts I and II also achieve higher similarity scores than HCI-LDA for EXINFMSRS (130%-180%) and show smaller gains for NASA SRS (3.13%-34.38%).

CTM yields lower similarity scores than HCI-LDA across all datasets, indicating comparatively weaker alignment with the original SRS documents. Overall, HCI-LDA achieves the highest similarity scores among the topic modeling approaches, while ChatGPT and human experts exhibit higher alignment in certain cases.

Human Experts vs. Generative AI (ChatGPT)

The comparison between human experts and ChatGPT highlights the effectiveness of generative AI in software requirements analysis. ChatGPT consistently achieves high similarity scores across all three SRS documents (0.55 for NASA, 0.42 for EXINFMSRS, 0.23 for SRS4.0), often surpassing or aligning closely with human experts.

NASA SRS: ChatGPT (0.55) outperforms Human Expert I (0.43) and Human Expert II (0.33).

ExinfmSRS: ChatGPT (0.42) is nearly equivalent to Human Expert II (0.43) and significantly higher than Human Expert I (0.23).

SRS4.0: ChatGPT (0.23) is comparable to Human Expert I (0.26) and Human Expert II (0.25).

How to cite this article:

Table 2 compares ChatGPT and the two human experts based on mean human performance (MHP) and ChatGPT Improvement Over MHP (CIM).

The MHP is the average performance of the two human experts, while the CIM is computed from the equation as follows:

$$\text{CIM} = \left(\frac{\text{CG} - \text{MHP}}{\text{MHP}} \right) \times 100 \quad (15)$$

Interpreters	SRS4.0	EXINFMSRS	NASA
ChatGPT	0.23	0.42	0.55
Human Expert I	0.26	0.23	0.43
Human Expert II	0.25	0.28	0.33
MHP	0.255	0.255	0.38
ChatGPT Improvement Over MHP (%)	-9.80%	64.71%	44.74%

TABLE 2: Comparative analysis of human expertise with generative AI

MHP, Mean Human Performance; SRS, Software Requirements Specification

The results indicate that ChatGPT achieves lower cosine similarity scores than human experts for SRS4.0 (-9.80%) but higher scores for EXINFMSRS (64.71%) and NASA SRS (44.74%). This suggests relatively lower lexical-semantic alignment with the original SRS4.0 document compared to the other datasets. A possible explanation may relate to the size and complexity of SRS4.0 (2305 words), which includes institution-specific terminology.

In contrast, higher similarity scores for EXINFMSRS (978 words) and NASA SRS (492 words) indicate closer alignment in these cases, potentially reflecting differences in document structure and terminology.

Human Expert vs. Topic Modeling (LDA, CTM)

The results indicate that human experts consistently outperform topic modeling approaches like LDA and CTM in similarity to SRS documents. Across all three cases:

- Human Expert I and Human Expert II exhibit moderate to high similarity with SRS documents, whereas LDA scores remain at 0.00 for all three.
- CTM achieves low similarity scores (e.g., 0.14 for NASA, 0.07 for EXINFMSRS, 0.22 for SRS4.0), confirming its limited effectiveness in structured document analysis.
- HCI-LDA outperforms CTM and LDA but still falls behind human experts in some cases.

A computation of Human Improvement Over LDA (in %) is derived from the MHP and the Mean LDA-Based Performance (MLP) to quantify the relative differences in cosine similarity scores across interpreters compared to HCI-LDA, highlighting variations in lexical-semantic alignment with the original SRS. The computation is defined in Equation (13).

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

$$\text{Human Improvement Over LDA} = \left(\frac{MHP - MLP}{MLP} \right) \times 100 \quad (16)$$

As can be observed from Table 3, results show that human experts perform better than LDA-based models in all cases, but the degree of improvement varies significantly depending on the dataset.

Interpreters	SRS4.0	EXINFMSRS	NASA
Human Expert I	0.26	0.23	0.43
Human Expert II	0.25	0.28	0.33
Mean Human Performance	0.255	0.255	0.38
LDA	0.00	0.00	0.00
HCI_LDA	0.52	0.10	0.32
CTM	0.22	0.07	0.14
Mean LDA-Based Performance	0.2467	0.0567	0.1533
Human Improvement Over LDA (%)	3.37%	350.10%	147.85%

TABLE 3: Analysis showing the relative effectiveness of different models in enhancing SRS interpretation beyond hierarchical clustering-based initialization

CTM, Correlated Topic Modeling; HCI-LDA, Hierarchical Clustering-based Latent Dirichlet Allocation; SRS, Software Requirements Specification

- i. In SRS4.0 (university management, 2305 words), human experts show a marginal improvement of 3.37% over LDA-based models, indicating relatively similar lexical-semantic alignment between both approaches for this dataset. This suggests that LDA-based models can achieve comparable alignment in structured documents where key themes are more uniformly distributed.
- ii. In EXINFMSRS (financial management, 978 words), human experts achieve higher cosine similarity scores than LDA-based models by 350.10%, indicating stronger alignment with the original SRS in this case.
- iii. In NASA (aerospace management, 492 words), human experts outperform LDA-based models by 147.85%, also reflecting higher lexical-semantic alignment with the reference document.

Overall, the results indicate variation in similarity-based alignment between LDA-based models and human experts across datasets.

Implications for industry, empirical software engineering, and software validation and verification

The findings of this study have significant implications for industry, empirical software engineering, and software validation and verification.

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

- i. In industry, the improved similarity scores of HCI_LDA highlights the potential of hierarchical clustering-based topic modeling for automating software requirements analysis, reducing manual effort, and enhancing the accuracy of software documentation interpretation. This can be particularly useful in large-scale software projects where understanding and managing requirements are critical for project success.
- ii. In empirical software engineering, the results provide evidence that clustering-based initialization can improve similarity-based alignment between generated representations and SRS documents, suggesting the value of incorporating more structure-aware initialization strategies in topic modeling approaches for software requirements analysis.
- iii. Lastly, for software validation and verification, the study highlights the importance of accurate requirement interpretation, as misinterpretations can lead to defective software specifications. By leveraging more context-aware and semantically enriched techniques, organizations can enhance the precision and reliability of software validation processes, ultimately leading to fewer defects and improved software quality.

Discussion

Experimental setup

HCI-LDA model was implemented on a personal computer equipped with an 11th Gen Intel® Core™ i5-1135G7 processor operating at 2.40 GHz (up to 2.42 GHz) and 16.0 GB RAM (15.7 GB usable). The system ran a 64-bit architecture (x64-based processor), providing sufficient computational capacity for tasks such as Word2Vec embedding generation, hierarchical clustering, and topic modeling using LDA. This hardware configuration ensured efficient memory management and high-performance processing for machine learning and NLP workloads.

The software environment was managed using Anaconda Navigator, which streamlined dependency management and package installation. All implementations were carried out in Python using Jupyter Notebook (v7.2.2). Key Python libraries included *Gensim* for LDA modeling, *Scikit-learn* for clustering, and NLTK and spaCy for text preprocessing. This integrated software stack provided a robust environment for implementing feature extraction, clustering, and topic modeling workflows.

The experiments utilized three SRS datasets. These datasets were analyzed through three approaches: human interpretation, ChatGPT interpretation, and topic modeling using LDA and CTM. The HCI-LDA model was implemented to enhance topic initialization and improve interpretability.

Library Importation

Pandas (*pd*) and *NumPy* (*np*) were used for structured data handling, enabling efficient storage, manipulation, and preprocessing of the SRS datasets. *Matplotlib* (*plt*) and *Seaborn* (*sns*) were utilized for visualizing clustering results, topic distributions, and word importance, improving interpretability.

To facilitate interactive topic visualization, the *pyLDAvis* and *pyLDAvis.gensim* libraries were employed, allowing for an intuitive exploration of topics and their relationships. Natural Language Toolkit was used for text preprocessing, including tokenization (*word_tokenize*) and stopwords removal (*stopwords*), ensuring that the dataset is cleaned before feature extraction. *Gensim*'s corpora and models are utilized for building the LDA model, which extracts meaningful topics from the processed text. Additionally, Word2Vec is used to generate dense vector representations of text, capturing semantic relationships and contextual meaning between words within the dataset.

Hierarchical clustering is implemented using *linkage* and *fcluster* from Scipy, enabling the initial grouping of SRS documents before applying LDA, ensuring a more structured topic modeling approach. This combination of preprocessing, clustering, and topic modeling techniques optimizes the performance and accuracy of the HCI-LDA model, making it well-suited for analyzing SRS documents.

LDA Implementation

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. *Cureus J Comput Sci* 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

The implementation of LDA topic modeling on SRS was done using two approaches: traditional LDA and HCI-LDA for improved topic initialization.

i. Traditional LDA Model Training: The LDA model (*lda_model*) is trained on the preprocessed corpus using 20 topics. The dictionary (*id2word*) links word IDs to their textual representation, and *passes=10* ensures multiple iterations for better topic coherence. A random seed (*random_state=42*) is set for reproducibility. Next step prints the extracted topics, each with 10 most important words, helping in topic interpretation.

ii. Hierarchical Clustering for LDA Initialization: The vectorization process extracts the words from the SRS dataset, converting texts to numerical vectors. Hierarchical clustering (*linkage_matrix*) groups similar sentences using the *ward* method, and *fcluster* assigns documents to one of the clusters. This ensures that semantically related words are grouped before topic modeling. A dictionary (*clustered_dict*) and corpus (*clustered_corpus*) are then created using clustered sentences, effectively structuring the input for LDA.

iii. LDA With Hierarchical Clustering Initialization: A second LDA model (*lda_hci_model*) is trained on the hierarchically clustered corpus, ensuring that topic formation starts from well-defined word groups. This improves coherence compared to traditional LDA. Topics discovered using this approach are then printed. This HCI-LDA approach enhances SRS topic modeling by leveraging hierarchical clustering for structured initialization.

Cosine Similarity Computation

i. Loading Datasets: The first line of code in the figure defines a dictionary (*file_paths*) containing the file paths for multiple CSV files for the three datasets and their respective six techniques each. These files store different versions of SRS processed with various topic modeling techniques. The dataset is then read into *pandas dataframes (df)* using *pd.read_csv()*, ensuring proper encoding to handle different character sets.

ii. Extracting Sentences for Similarity Analysis: A list (*sentences*) is created by extracting the first sentence from each dataset. This step helps in summarizing the datasets for textual similarity comparison, allowing a quick overview of the most representative text from each file.

iii. Vectorization: Word2vec is initialized and used to transform the text data into numerical feature vectors. This step helps in converting textual data into a format suitable for computing similarity between requirement documents.

iv. Computing Cosine Similarity: The cosine similarity matrix (*cosine_sim*) is computed using *cosine_similarity()*. This step helps measure the degree of similarity between different datasets based on their vector representations, highlighting textual resemblance among various versions of CSV data instances.

v. Creating and Visualizing a Heatmap: A heatmap visualization is generated using *seaborn.heatmap()* to present the cosine similarity values between datasets. The *annot=True* argument ensures that the similarity scores are displayed in the heatmap, and *cmap='coolwarm'* enhances the colour contrast for better readability.

Conclusions

This study demonstrates that integrating hierarchical clustering-based word initialization into LDA improves the lexical-semantic alignment between generated representations and SRS documents. HCI-LDA consistently outperformed LDA and CTM, achieving higher similarity scores and coherence, indicating closer textual correspondence to the original SRS documents. ChatGPT also showed strong performance, in some cases surpassing human experts in similarity scores, particularly in specialized domains. Overall, the findings suggest that HCI-LDA is an effective method for structured topic modeling in SRS documents under similarity-based evaluation and highlight the potential of combining clustering-based initialization with generative AI for enhanced representation of SRS documents.

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Michael Olusanya, Oluwasefunmi Arogundade, Dada O. Aborisade, Musiliu Raji, Oluwafolake Ojo

Acquisition, analysis, or interpretation of data: Michael Olusanya, Oluwasefunmi Arogundade, Dada O. Aborisade, Musiliu Raji

Drafting of the manuscript: Michael Olusanya

Critical review of the manuscript for important intellectual content: Oluwasefunmi Arogundade, Dada O. Aborisade, Musiliu Raji, Oluwafolake Ojo

Supervision: Oluwasefunmi Arogundade, Dada O. Aborisade, Musiliu Raji

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Beyersdorffer P, Münch J, Kuhrmann M: [Alignment of quality models for assessing software requirements in large-scale projects: a case from space](#). 2023 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Edinburgh, United Kingdom. 2023, 1-9. [10.1109/ICE/ITMC58018.2023.10332295](#)
2. Dar HS, Imtiaz S, Lali MI: [Reducing requirements ambiguity via gamification: comparison with traditional techniques](#). Computational Intelligence and Neuroscience. 2022, 2022:3183411. [10.1155/2022/3183411](#)
3. Fucci D, Alégroth E, Axelsson T: [When traceability goes awry: An industrial experience report](#). Journal of Systems and Software. 2022, 192:111389. [10.1016/j.jss.2022.111389](#)
4. Tiun S, Mokhtar UA, Bakar SH, Saad S: [Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text](#). Journal of Physics: Conference Series. 2020, 1529:042077. [10.1088/1742-6596/1529/4/042077](#)
5. Ndlela M, Tanner M: [Business analysts' contributions to the dynamic capabilities of agile software development teams](#). Information Technology & People. 2023, 36:1-20. [10.1108/itp-08-2021-0656](#)
6. Qamar N, Sabahat N, Mosavi A: [Evaluating the impact of pair documentation on requirements quality in agile software development](#). IEEE Access. 2025, 13:45784-45794. [10.1109/access.2025.3550123](#)
7. Rahimi N, Eassa F, Elrefaei L: [One- and two-phase software requirement classification using ensemble deep learning](#). Entropy. 2021, 23:1264. [10.3390/e23101264](#)
8. Ahmad I, Rasheed U, Khalil I, Anwaar Z, Butt WH: [Automated software requirements prioritization using natural language processing](#). 2025 6th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan. 2025. 1-7. [10.1109/ICACS64902.2025.10937796](#)
9. Ahmad A, Feng C, Khan M, Khan A, Ullah A, Nazir S, Tahir A: [A systematic literature review on using machine learning algorithms for software requirements identification on stack overflow](#). Security and Communication Networks. 2020, 2020:8830683. [10.1155/2020/8830683](#)

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>

10. Silva CC, Galster M, Gilson F: [Topic modeling in software engineering research](#). Empirical Software Engineering. 2021, 26:120. [10.1007/s10664-021-10026-0](#)
11. Dakshi KG, Yue X, Yuefeng L: [A semantics-enhanced topic modelling technique: semantic-LDA](#). ACM Transactions on Knowledge Discovery from Data. 2024, 18:1-27. [10.1145/3639409](#)
12. Gao Q, Jiang M: [Exploring technology fusion by combining latent Dirichlet allocation with Doc2vec: a case of digital medicine and machine learning](#). Scientometrics. 2024, 129:4043-4070. [10.1007/s11192-024-05069-1](#)
13. Rieger J, Jentsch C, Rahnenführer J: [LDAPrototype: a model selection algorithm to improve reliability of latent Dirichlet allocation](#). PeerJ Computer Science. 2024, 10:e2279. [10.7717/peerj-cs.2279](#)
14. Saklamaeva V, Pavlič L: [The potential of AI-driven assistants in scaled agile software development](#). Applied Sciences. 2024, 14:319. [10.3390/app14010319](#)
15. Mohamed KA, Din J, Baharom S: [A tool to detect pragmatic ambiguity with possible interpretations suggestion in software requirement specifications](#). International Journal of Synergy in Engineering and Technology. 2022, 3:52-60.
16. Darnoto BRP, Siahaan D: [MultiPhiLDA for detection irrelevant software requirement specification](#). 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Banyuwangi, Indonesia. 2021. 92-97. [10.1109/ICOMITEE53461.2021.9650315](#)
17. Oran AC, Santos G, Gadelha B, Conte T: [A framework for evaluating and improving requirements specifications based on the developers and testers perspective](#). Requirements Engineering. 2021, 26:481-508. [10.1007/s00766-021-00352-6](#)
18. Mucahit C, Savas Y, Ayşe B: [Natural language processing for software requirement specifications](#). 31st Annual International Conference on Computer Science and Software Engineering (CASCON '21). 2021. 308-309.
19. Younas M, Jawawi DN, Ghani I, Shah MA: [Extraction of non-functional requirement using semantic similarity distance](#). Neural Computing and Applications. 2020, 32:7383-7397. [10.1007/s00521-019-04226-5](#)
20. Ekinci E, Omurca SI: [NET-LDA: a novel topic modeling method based on semantic document similarity](#). Turkish Journal of Electrical Engineering and Computer Sciences. 2020, 28:2244-2260. [10.3906/elk-1912-62](#)
21. Priyadi Y, Djunaidy A, Siahaan D: [Requirements dependency graph modeling on software requirements specification using text analysis](#). 2019 1st International Conference on Cybernetics and Intelligent System (ICORIS), Denpasar, Indonesia. 2019. 221-226. [10.1109/ICORIS.2019.8874920](#)

How to cite this article:

Olusanya M, Arogundade O, Aborisade D O, et al. (April 22, 2026) Data-Driven Initialization in Topic Modeling for Enhanced Software Requirements Specification Interpretation. Cureus J Comput Sci 3 : es44389-025-00054-y. DOI <https://doi.org/10.7759/s44389-025-00054-y>