Exploring Manifold-Based Clustering Techniques for Enhanced Inductive Thematic Analysis

Jesus A. Beltran[1(✉)], Hanna Mofid[2], Harita Parikh[1], Jaydeep Gondaliya[1], Diego Guzman[2], Jenil Shah[1], Lizbeth Escobedo[3], and Franceli Cibrian[4]

[1] California State University, Los Angeles, USA
{abeltr99,hparikh5,jgondal,jshah25}@calstatela.edu
[2] University of California, Irvine, USA
{hmofid,guzmand7}@uci.edu
[3] Dalhousie University, Halifax, NS, Canada
lizbeth.escobedo@dal.ca
[4] Fowler School of Engineering, Chapman University, Orange, USA
fcibrian@chapman.edu

Abstract. In this paper, we introduce Augmented Thematic Analysis with Large Language Models (ATA-LLM), a novel framework that integrates manifold learning algorithms and clustering techniques to support inductive thematic analysis. This qualitative method, widely used in software engineering, is essential for uncovering patterns and understanding human factors and software requirements. Traditional thematic analysis involves data coding, theme identification, and the interpretation of complex narratives, making it a labor-intensive and time-consuming process. Recent advances in large language models (LLMs) offer promising opportunities; however, it remains unclear how comparable these approaches are to traditional human thematic analysis. To address this gap, we evaluated ATA-LLM using a validated qualitative dataset and compared the outcomes against human-coded analysis. Our findings indicate that within the ATA-LLM framework, DenseMAP and UMAP effectively preserve both local and global structures of high-dimensional data, resulting in more coherent and meaningful themes than other techniques. These results highlight the potential of ATA-LLM to enhance the rigor, consistency, and efficiency of inductive thematic analysis.

Keywords: Inductive Thematic Analysis · Large Language Models · Clustering

1 Introduction

Thematic Analysis (TA) is a method used in software engineering to analyze qualitative data and uncover patterns related to technical practices and human-centered aspects of software development [21,36]. TA is beneficial for understanding complex phenomena and uncovering software requirements [21,36]. TA involves systematically reviewing textual data (often collected from interviews,

S. Hussain et al. (Eds.): SEET 2025, CCIS 2725, pp. 464–480, 2026.
https://doi.org/10.1007/978-3-032-08977-9_29

focus groups, observations, or open-ended surveys) and identifying "codes" that represent thoughts, ideas, or attitudes. Those codes are iteratively grouped into broader categories or higher-level "themes" that help understand insights into social, technical, and organizational processes [46].

Depending on the research or team goals, TA can follow either a deductive or inductive approach [7]. Deductive TA starts with a predefined framework or theory, while inductive TA allows themes to emerge from the data without prior assumptions. Inductive TA typically involves six phases: (1) familiarization with the data, (2) generating initial codes, (3) identifying candidate themes, (4) reviewing themes, (5) defining and naming themes, and (6) producing the final analysis and themes [6]. Although TA provides an in-depth understanding, it tends to be labor-intensive and time-consuming, and the interpretative nature of TA requires human involvement and teamwork to ensure the validity of the findings [7]. The burden could be intensified with a large amount of data.

Recent progress in generative Large Language Models (LLMs) offers a promising avenue for addressing challenges and enhancing thematic analysis techniques due to their advanced comprehension capabilities in software engineering [11,26]. Some of the potential advantages are: increase efficiency and scalability [37,50]; augmenting human cognitive and reasoning abilities by leveraging LLMs to reduce cognitive load and enable deeper text analysis [26]; identifying novel themes and patterns, especially in large and complex text corpora [37].

LLMs' methods aim to identify recurring patterns in text data [26]. They rely on representing the data in numerical form. Using text embeddings, a manifold learning technique, and a clustering algorithm, underlying themes are revealed. The current research trajectory in integrating LLMs with qualitative analysis has focused primarily on deductive analysis [10,51]. Initial strides have also been taken to explore the application of LLMs to inductive analysis, indicating the viability of utilizing these models to support thematic analysis [12]. However, it is unclear how comparable these approaches are to the original human-coded analysis. In this sense, our research question is as follows:

Which combination of manifold learning and clustering techniques most effectively supports inductive thematic analysis of semi-structured interview data in comparison with human-coded analysis?

To answer this question, this paper presents a systematic empirical comparison of leading manifold learning algorithms combined with clustering techniques within the context of inductive thematic analysis included in Augmented Thematic Analysis with Large Language Model (ATA-LLM), a framework designed to leverage the capabilities of LLMs to enhance thematic coding, pattern recognition, and insight generation while maintaining the interpretive depth and reliability of human analysis. Our findings show that within the ATA-LLM framework, DenseMAP and UMAP, two non-linear dimensionality reduction algorithms that preserve both local and global structures of high-dimensional data, capturing semantic relationships, generated more coherent and meaningful themes than other techniques. These results highlight the potential of ATA-LLM to enhance the rigor, consistency, and efficiency of inductive thematic analysis.

2 Background and Related Work

2.1 The Role of Thematic Analysis in Software Engineering

Qualitative methods have been used to gain deep insights into how people think, feel, and behave. In software engineering, these methods are increasingly recognized as essential for understanding the complex, human-centered dimensions of software development processes and practices [15,48]. Among the most frequently used qualitative methods in software engineering are Grounded Theory [20] and Thematic Analysis (TA) [28], both of which enable software engineers to derive rich, contextualized insights from textual data, mostly from interviews and observations [1,13].

Software engineers have adopted and adapted qualitative methods over the past two decades. For instance, Seaman and Easterbrook et al. found that software engineering used qualitative analysis to analyze interviews, ethnography, and participant observation [16,48]. Later work focused on developing adaptations of those techniques, but focused on the context of software systems engineering. For example, Runeson et al. provided detailed methodological guidance on case study research [42,43]. Sharp et al. explored the role of ethnographic methods in understanding software teams and organizational settings [49]. More recently, Hoda has proposed a socio-technical adaptation of Grounded Theory tailored to software engineering challenges, emphasizing the importance of aligning methods with the unique socio-technical realities of software development [21,22]. In a similar manner, Lenberg et al. [28] argue that such adaptations represent a critical step forward in legitimizing and strengthening the use of qualitative research in software engineering.

Despite these advances, there is still uncertainty about whether existing methodological guidelines are sufficient to fully support the application of qualitative methods in software engineering practice [47]. This question has become especially urgent with the rise of AI-enabled tools for qualitative data analysis [3,22]. As Seaman observes, these emerging technologies offer "rich opportunities to test and explore the boundaries of what is possible and wise" in qualitative research [47]. However, integrating AI tools into rigorous, theory-driven analysis remains an open challenge, requiring new guidance specific to the epistemological and practical demands of the software engineering discipline.

2.2 Integrating AI Into Thematic Analysis

The use of AI to support TA has increased, particularly given the improvements of algorithms in natural language processing (NLP) and LLMs. Early work combined manual coding with tools like LIWC, which quantifies word categories across text to support interpretation [35,39]. Gauthier et al. [18] introduced a toolkit to aid in data visualization and filtering during thematic analysis, though it lacked automated theme identification.

LLMs are now applied primarily in deductive thematic analysis—linking predefined codes to qualitative data. Studies show promising results: Xiao et al. [51]

achieved moderate agreement between GPT-3 and human coders (Cohen's K = 0.61), and Chew et al. [10] reported even higher agreement (above 0.76) with GPT-3.5. Exploration into inductive thematic analysis using LLMs is still in the early stages. De Coster et al. [12] demonstrated that GPT-3.5 can approximate Braun and Clarke's six-step framework, generating themes from data; however, further validation is needed. The quality of AI-generated themes is evaluated using metrics such as inter-coder agreement (e.g., Cohen's Kappa), precision, recall, F1-score, and topic coherence measures like NPMI and UCI scores [11, 19, 25]. However, human validation remains crucial for assessing interpretability, accuracy, and ethical considerations [34, 50].

Despite promising advances, AI-assisted thematic analysis still faces critical technical challenges, particularly in managing and interpreting the high-dimensional text embeddings produced by LLMs. These embeddings are foundational for identifying and clustering semantically similar data segments, yet their complexity can hinder analysis and theme discovery without proper preprocessing. To address this, our work explores two key components essential for scaling and refining AI-driven thematic analysis in software engineering: (1) high-dimensional data representation and dimensionality reduction techniques to improve interpretability, and (2) clustering algorithms to support robust, inductive theme identification. The following sections detail our methodological approach and empirical evaluation of these techniques.

2.3 High-Dimensional Data Representation and Dimensional Reduction Algorithms for Thematic Analysis

Thematic analysis using AI techniques increasingly relies on transforming qualitative textual data into numerical representations, most commonly through text embeddings generated by LLMs, which enables clustering and finding patterns across text ("themes"). These embeddings, often high-dimensional, encode semantic and contextual meaning but pose interpretability and computational challenges. As a result, dimensionality reduction techniques are employed to enhance computational efficiency while preserving semantic integrity in qualitative research [38]. Principal Component Analysis (PCA) has been widely used to reduce embedding dimensions while maintaining variance in textual themes, as seen in studies applying LLM-based thematic profiling [24].

Similarly, Uniform Manifold Approximation and Projection (UMAP) has demonstrated effectiveness in non-linear feature reduction, enabling researchers to visualize and cluster emergent themes from LLM-generated embeddings [17]. The integration of t-SNE (t-Distributed Stochastic Neighbor Embedding) has also been noted in studies that seek to improve interpretability when mapping high-dimensional textual data into low-dimensional spaces for qualitative coding validation [19]. Furthermore, sentence embeddings derived from transformer models such as BERT and GPT-4 have been subjected to cosine similarity-based clustering, refining theme extraction through vector space modeling [12, 25]. One of the notable approaches is the use of BERT-based embeddings, where text is

converted into 768-dimensional dense vectors that encapsulate both word semantics and context [8]. In the study "BERT-Based Deep Embedded Clustering for Topic Modeling" [8], BERT embeddings were combined with Deep Embedded Clustering (DEC) and Improved Deep Embedded Clustering (IDEC) to reduce dimensionality and optimize the clustering process simultaneously. This approach preserves semantic coherence and improves theme identification in high-dimensional data. The study "Opinion Text Clustering Using Manifold Learning Based on Sentiment and Semantics Analysis" [23] employed Doc2Vec embeddings to convert opinion texts into 300-dimensional vectors. To overcome the curse of dimensionality, the ISOMAP manifold learning algorithm was applied to reduce the text representations while preserving the semantic structure [23]. Overall, these methods enhance computational efficiency and data visualization, but challenges remain in selecting optimal reduction parameters that maintain thematic consistency without information loss.

2.4 Clustering Algorithm for Thematic Analysis

Applying clustering algorithms in TA has significantly advanced with the integration of LLMs, enabling efficient organization of text-based data into meaningful themes [11]. Traditional hierarchical clustering methods, such as agglomerative clustering, have been employed to structure qualitative data by merging semantically similar text segments iteratively [24]. Meanwhile, K-Means clustering, widely used in NLP tasks, has demonstrated efficacy in grouping LLM-generated embeddings into distinct thematic clusters, particularly when combined with word embeddings from transformer models like BERT and GPT-4 [25]. Studies exploring DBSCAN (Density-Based Spatial Clustering of Applications with Noise) have highlighted its ability to identify outlier topics and rare themes, offering a robust framework for analyzing highly variable qualitative datasets [19,26].

More recent advancements involve topic modeling approaches, such as Latent Dirichlet Allocation (LDA) and Structural Topic Modeling (STM), which enable probabilistic clustering of text into coherent topics for further manual refinement [12,17]. Additionally, cosine similarity-based clustering has been leveraged in thematic analysis to compute semantic distances between textual embeddings, facilitating automated theme detection with high interpretability [45]. The use of Deep Embedded Clustering (DEC) and Improved DEC (IDEC) in BERT-based clustering [8] allows for direct clustering in the low-dimensional latent space. Similarly, K-Means clustering applied to ISOMAP-reduced embeddings [23] effectively groups texts based on sentiment and semantics. This body of work shows that clustering methods improve efficiency in theme extraction and organization; however, challenges persist in optimizing the number of clusters and refining model interpretability to ensure thematic coherence in qualitative research.

Overall, thematic analysis is widely used in software engineering, however, its manual implementation remains time-consuming and cognitively demanding. Recent applications of LLMs have focused mainly on deductive coding. To

our knowledge, there is limited validation and comparison with a human-coded system, especially for inductive coding. Although techniques like UMAP and DenseMAP offer promise to be used in qualitative analysis, their integration into inductive thematic analysis has not been explored. This paper addresses this gap by introducing and evaluating ATA-LLM, a framework that combines LLMs, dimensionality reduction, and clustering to enhance the coherence, efficiency, and interpretability of inductive thematic analysis in software engineering.

3 Methodology

We conducted a study to evaluate the outcome of the ATA-LLM compared to a human-generated codebook. We used an open dataset from the Researching Students' Information Choices (RSIC) study [9], which we used as a ground truth. ATA-LLM generates a codebook using five steps (see Fig. 1). First, each transcript excerpt and its surrounding context are converted into open codes. Second, the open code set is reduced to keep the unique code or those with low semantic similarity. Third, the dimensionality of embedding is reduced. Then, the resulting embeddings are clustered to organize unique open code into potential organic themes. Lastly, a theme is generated using LLM for each cluster.

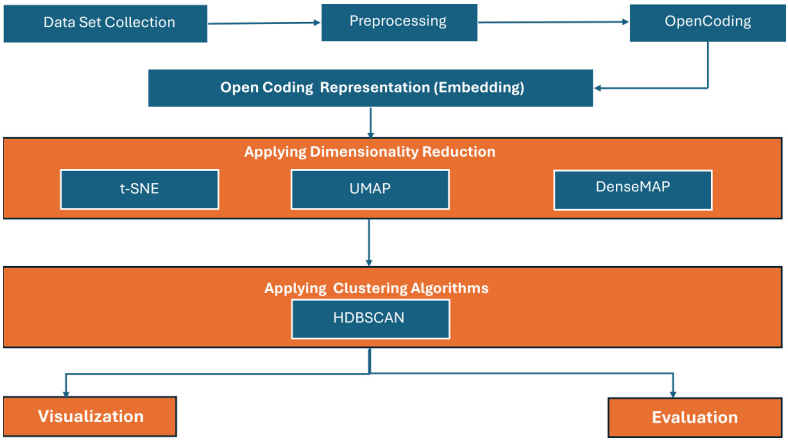


Fig. 1. The overall scheme of the Augmented Thematic Analysis - Large Language Model (ATA-LLM) approach.

3.1 Dataset: Transcripts and Codebook

The dataset [9], from the Researching Students' Information Choices (RSIC) study, examines how students assess and select online information sources. Research in the dataset aims to understand students' decision-making processes when evaluating the credibility and relevance of digital resources during

a science-related research task. It includes qualitative and quantitative data, providing insights into how students from different educational levels interact with search engine results. The dataset comprises 1,201 questionnaire responses, 175 interview transcripts, 175 simulation task decisions, and 175 think-aloud transcripts. Data was collected from participants across six educational stages: elementary school, middle school, high school, community college, undergraduate, and graduate levels. The study setting included simulated online searches designed to replicate real-world research behaviors.

This dataset's key component is providing a codebook (i.e., the set of codes, categories, relationships, definitions, and examples [44]). This codebook is traditionally used to enhance consistency in analysis, supporting methodological transparency and reproducibility [40].

3.2 Preprocessing: Open Code Generation and Embedding

In ATA-LLM, the preprocessing phase begins by segmenting interview transcripts into chunks that include each sentence and its surrounding context. For each segment, a pre-trained LLM is prompted to generate open codes, i.e., short phrases (typically 1 to 4 words) that capture the core meaning of the text. These open codes serve as the foundational units for thematic analysis. Figure 2 illustrates the prompt used for open code generation. Each open code is then transformed into an embedding, a high-dimensional vector representation, such that semantically similar codes are positioned closer together in the embedding space. The code generation process was guided by a research question similar to that used in [9], ensuring consistency across the dataset.

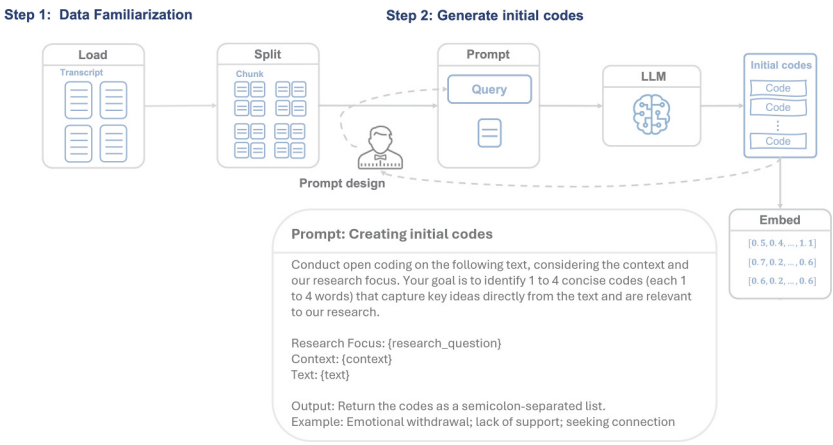


Fig. 2. Overview of the preprocessing pipeline, open code generation, and embedding workflow used for inductive thematic analysis.

3.3 Open Code Embedding Dimensionality Reduction

Open-code embeddings are represented in high dimensionality. For instance, in the embedding model 'text-embedding-ada-002', each text segment is mapped to a 1536-dimensional vector. Organizing code into meaningful groups to identify patterns can be difficult in high-dimensional spaces due to the curse of dimensionality [5]. For this reason, we apply three well-known manifold algorithms to reduce the dimensionality of each open-code embedding. The methods include:

- UMAP (Uniform Manifold Approximation and Projection) [32]
- DenseMAP [33]
- t-SNE (t-Distributed Stochastic Neighbor Embedding) [29]

3.4 Theme Identification Through Clustering Algorithms

After generating the open codes, the next step is to group them based on similarity. These grouped codes will form potential themes, which can vary in shape and size. Given the properties of these groups, we have chosen to use the clustering algorithm, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [31]. This unsupervised machine learning model effectively identifies meaningful clusters with varying densities while filtering out noise.

Then, a potential theme for each cluster was inferred via GPT-4o-mini. We constructed our prompt by providing only the initial code per cluster:

Prompt: Themes inferred.

You are a qualitative researcher. You are helping me write a potential theme or category for initial codes.
Write a potential category name or topic using the following list of topics.
INITIAL CODES:{codes} THEME TITLE:"

3.5 Evaluation Criteria

The evaluation criteria include unsupervised and supervised metrics, which help determine the optimal combination of the dimensionality reduction algorithm and the clustering algorithm for thematic analysis.

The unsupervised metrics help evaluate the theme clustering quality when a human-annotated codebook is unavailable. For this purpose, we employ the silhouette score (S) [41] to evaluate cluster separation, topic coherence (TC) [27] to measure the semantic similarity of the top open codes within each theme, and topic diversity (TD) [14] to quantify the uniqueness of code vocabularies across the model.

The supervised metrics allow us to evaluate the themes generated by the ATA-LLM with a human-annotated codebook. For this purpose, we employ a

similar approach to [30], in which we embed both the human-annotated codebook and the ATA-LLM-generated themes into a shared semantic space using the embedding model. Then, we compute the average pairwise cosine similarity between aligned themes. The measure range is $[0, 1]$, where 1 indicates a perfect similarity.

4 Experiments and Results

4.1 Parameter Settings and Hyperparameter Optimization

Each manifold and clustering algorithm has symbolic and numeric parameters. To find the best configuration over the hyperparameter search space Θ, we model this problem as a black-box optimization problem in which, given a specific pipeline configuration in ATA-LLM ($\theta \in \Theta$), we evaluate the quality using the following utility function U:

$$U(\theta) = \lambda_1 TC(\theta) + \lambda_2 TD(\theta) + \lambda_3 S(\theta)$$

where λ_1, λ_2, and λ_3 are weights used in the utility function to set the preference over metrics TC, TD, and S. In our experiments, we used $\lambda_1 = 0.3$, $\lambda_2 = 0.6$, and $\lambda_3 = 0.1$, respectively. Then, the tuning problem can be stated as:

$$\theta^* = \arg\max_{\theta \in \Theta} U(\theta)$$

To solve this tuning problem efficiently, we employed the open-source framework Optuna [2], which supports the search for hyperparameters by using Bayesian optimization. Table 1 presents the hyperparameters and their corresponding values. Over 200 trials were conducted to find the best hyperparameter configuration.

4.2 Results

Using the optimal hyperparameters found by Optuna (see Table 2), we evaluated each configuration based on three metrics: topic coherence (TC), topic diversity (TD), and silhouette score (S). The results are presented in Table 3. From Table 3, we can observe that UMAP and DenseMAP have high topic coherence ($TC = 0.893$), indicating a strong consistency in the themes generated. Conversely, the combination of t-SNE and HDBSCAN demonstrates the highest topic diversity ($TD = 0.975$). Overall, all three manifold algorithms demonstrate a competitive topic diversity score, suggesting that ATA-LLM can effectively name themes using varied vocabulary within clusters. Concerning the silhouette score (S), which evaluates cluster separation, DenseMAP receives the highest rating ($S = 0.152$).

Table 1. Hyperparameters used in Optuna optimization [2].

Algorithm	Hyperparameter	Hyperparameter values
Dimensional Reduction Algorithm		
UMAP	n_neighbors	[5, 50]
	min_dist	[0.0, 0.5]
	metric	cosine
DenseMAP	n_neighbors	[5, 50]
	min_dist	[0.0, 0.5]
	dens_lambda	[0.1, 0.3]
tSNE	perplexity	[5, 50]
	learning_rate	auto
Clustering Algorithm		
HDBSCAN	min_cluster_size	[5, 100]
	min_samples	[5, 20]
	cluster_selection_epsilon	[0.1, 1.0]

Table 2. Best hyperparameter configuration identified by Optuna [2]

Algorithm	Hyperparameter	Hyperparameter values
UMAP	n_neighbors	21
	min_dist	0.10
HDBSCAN	min_cluster_size	23
	min_samples	10
	cluster_selection_epsilon	0.229
DenseMAP	n_neighbors	14
	min_dist	0.25
	dens_lambda	2.79
HDBSCAN	min_cluster_size	15
	min_samples	3
	cluster_selection_epsilon	0.331
tSNE	perplexity	11
HDBSCAN	min_cluster_size	44
	min_samples	17
	cluster_selection_epsilon	0.919

Table 3. Comparison of metrics across different dimensionality reduction and clustering configurations for thematic analysis.

Metric	UMAP + HDBSCAN	DenseMAP + HDBSCAN	t-SNE + HDBSCAN
TC	**0.893**	**0.893**	0.817
TD	0.935	0.925	**0.975**
S	0.140	**0.152**	−0.046

Open Code Embedding Dimensionality Reduction. Our ATA-LLM framework identified 4,536 initial codes from semi-structured interviews [9]. To capture the semantic relationships among these codes, we generated 1,536-dimensional embedding vectors for each code (see Sect. 3.2). We then applied three manifold learning algorithms for dimensionality reduction.

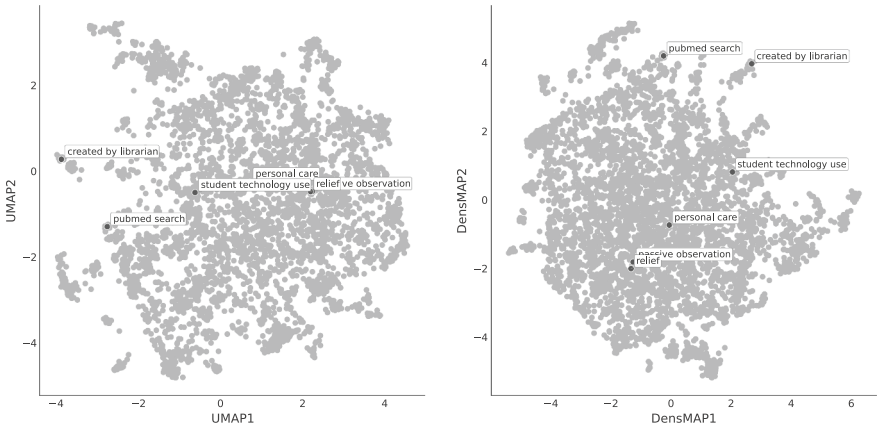
Figure 3 presents the reduced embeddings of a sample set of six open codes. Notably, the codes "pubmed search" and "created by librarian" consistently appear nearby across all three manifold algorithms, suggesting substantial semantic similarity. In contrast, we also observe several initial codes positioned far from the dense cloud of points. We hypothesize that these outliers either represent semantically unique codes, lack sufficient transcript data to support them, or align with other initial codes.

Grouping Codes Into Themes: We employed the unsupervised clustering technique HDBSCAN [31] to identify distinct thematic groups within the initial codes. Each code was treated as mutually exclusive (i.e., they only belong to one category). To improve the quality of the clusters, we pre-processed the data by removing outlying codes that HDBSCAN identified. This approach is similar to affinity diagramming, where ideas are organized into groups, although some outlier ideas should not be forced into categories.

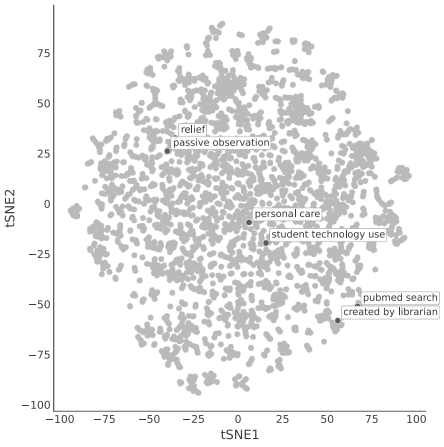
Figure 4 shows the cluster assignments produced by HDBSCAN [31] for each manifold learning algorithm: UMAP, DenseMAP, and t-SNE. Across all three methods, HDBSCAN consistently identifies four well-separated clusters, providing a clear structure for theme inference. In addition, Fig. 4 shows the potential theme for each cluster, inferred using GPT-4o-mini. Notably, UMAP and DenseMAP are highly comparable thematic results, with three out of four inferred themes that align closely. Both algorithms share two of four themes with t-SNE (see Table 4).

Table 4. Similar thematic labels identified across UMAP, DenseMAP, and t-SNE projections

UMAP	DenseMAP	t-SNE
Evaluating Credibility and Uncertainty in Online Research Sources	Challenges of Online Information Sourcing in Research Tasks	Evaluation and Credibility of Online Sources
Citation Management Confusion and Tools	Evaluation and Preference for Reference Management Tools	Citation Management and Tool Utilization
Perceptions of Wikipedia's Credibility and Reliability as a Source	Reliability and Credibility of Wikipedia as a Source	



(a) UMAP (b) DenseMAP

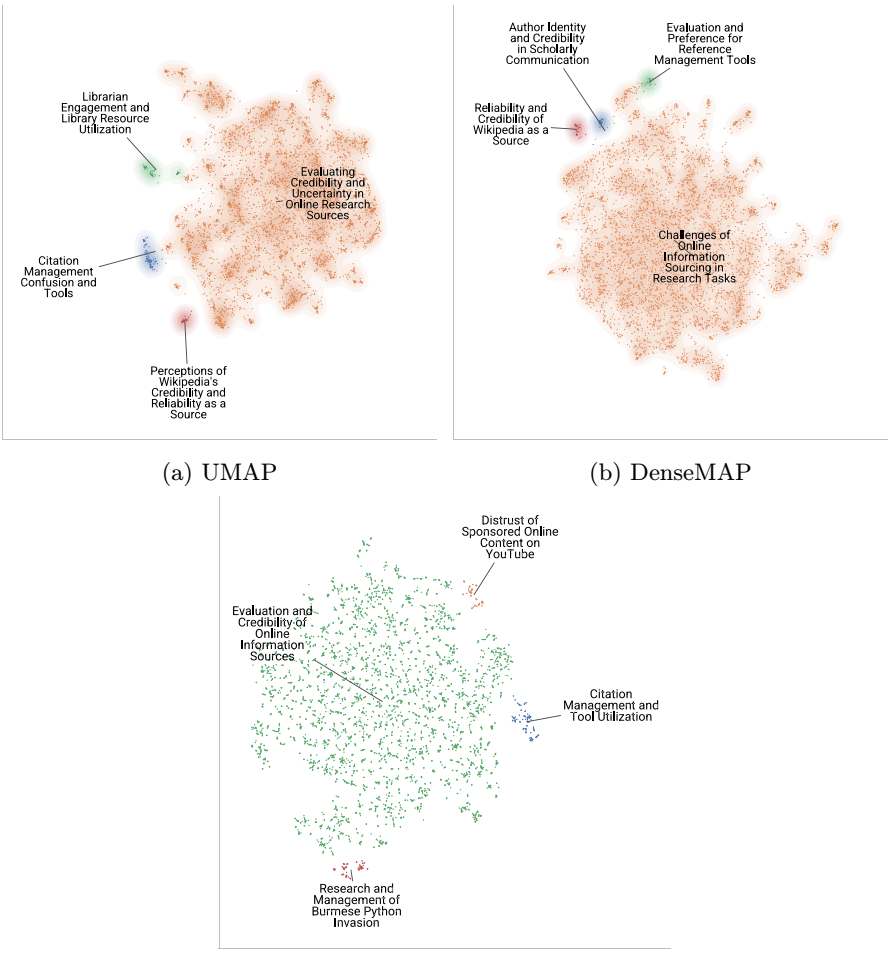


(c) t-SNE

Fig. 3. Dimensionality reduction of open code embeddings from the dataset in [9] using three different manifold learning algorithms. Blue dots represent a random sample of open codes, where spatial proximity indicates semantic similarity. (Color figure online)

Evaluation Between Human and ATA-LLM-Generated Themes. To evaluate the similarity between the themes generated by the ATA-LLM and a human-annotated codebook, we employ a similar approach to [30], in which we embed both the human-annotated codebook and the ATA-LLM-generated themes into a shared semantic space using an embedding model. We then compute the average pairwise cosine similarity between aligned themes. The measure ranges from 0 to 1, where 1 indicates a perfect similarity.

The human-annotated codebook in [9] presents one theme and four levels of subthemes: Theme (I, II, III), Subtheme (A, B, C), Subtheme (1, 2, 3), Subtheme



(a) UMAP (b) DenseMAP

(c) t-SNE

Fig. 4. Theme Identification through Clustering Algorithms.

(a, b, c), Subtheme (i, ii, iii). For this reason, we compute the average cosine similarity between human-generated themes and those generated by our three different approaches for each level.

The subtheme level (A, B, C) in the human-annotated codebook [9] exhibited the highest average cosine similarity. DenseMAP produces the highest average score at this level, with a similarity of 0.77. Similarly, UMAP and t-SNE show a similarity score of 0.76.

Overall, all three manifold learning approaches demonstrated reliable performance across subtheme granularities, with DenseMAP providing slightly more stable scores across all theme structures.

5 Conclusions, Limitations, and Future Perspectives

This study evaluated the effectiveness of integrating manifold learning algorithms with HDBSCAN to support inductive thematic analysis using LLMs. Our findings show that DenseMAP [33] combined with HDBSCAN [31] offers a more substantial alignment with human-coded themes. We selected these techniques to capture semantically coherent and contextually meaningful themes. In contrast, some manifold learning algorithms, such as t-SNE, failed to preserve the nuance of qualitative data, ultimately oversimplifying or distorting the semantic relationships among codes. These limitations are consistent with previous findings that t-SNE can distort global structure and exaggerate local patterns, making it less reliable for interpretable clustering tasks [29].

A key insight from our study is the importance of balancing automation with human supervision. While specific sections of the ATA-LLM pipeline, such as the open code embedding (see Sect. 3.3) and dimensionality reduction, can be automated without significantly compromising interpretability. However, other stages, such as open coding and theme identification, benefit substantially from human-in-the-loop involvement to ensure contextual accuracy and to mitigate potential biases. Integrating domain expertise at these stages allows for more contextually accurate theme refinement, such as merging semantically similar clusters or splitting broad clusters into more meaningful subthemes. Moreover, incorporating human feedback into the LLM-based coding process helps ensure that the resulting themes align with the underlying research questions and analytical objectives. Our comparative analysis also reveals that LLM-driven approaches frequently generate codes and themes with uniform granularity. In contrast, human coders adjust the level of abstraction according to context and interpretive judgment. Supporting user control over granularity and consistency can improve the usability and trustworthiness of AI-assisted thematic analysis tools.

Some limitations of our study include that our evaluation is based on a single dataset, one datatype, and tested with only one large language model (GPT-4o-mini). These constraints may limit generalizability, and we acknowledge that performance will likely vary across domains, datasets, and model types. Nonetheless, we view this case study as a foundation for designing future AI-based tools that ethically and transparently support inductive qualitative analysis.

Ethical considerations must be central in the development of these types of tools. It is known that LLMs are trained on large, generalized corpora that may encode biases [4], which may influence how themes are identified and interpreted. Therefore, to increase trust in AI-assisted qualitative analysis, we advocate for augmented approaches that prioritize human judgment and emphasize transparency and reproducibility. This can also bridge the gap between qualitative and quantitative research communities by showing how rigorous and replicable qualitative methods can benefit from AI support. Educators must also prepare students to use these AI tools ethically by building a deep understanding of their potential and limitations among students.

Future work should explore human-in-the-loop AI-assisted tools that can serve as collaborative partners, enabling scalable, interpretable, and ethically grounded thematic analysis. Such tools should aim to preserve the richness of human insight while leveraging the efficiency and scalability of machine learning.

References

1. Adolph, S., Kruchten, P., Hall, W.: Reconciling perspectives: a grounded theory of how people manage the process of software development. J. Syst. Softw. **85**(6), 1269–1286 (2012)
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
3. Bano, M., Hoda, R., Zowghi, D., Treude, C.: Large language models for qualitative research in software engineering: exploring opportunities and challenges. Autom. Softw. Eng. **31**(1), 8 (2024)
4. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: can language models be too big?. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 610–623 (2021)
5. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Beeri, C., Buneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49257-7_15
6. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**(2), 77 (2006)
7. Braun, V., Clarke, V.: Toward good practice in thematic analysis: avoiding common problems and be (com)ing a knowing researcher. Int. J. Transgender Health **24**(1), 1–6 (2023)
8. Cahyadi, D.J., Murfi, H., Satria, Y., Abdullah, S., Widyaningsih, Y.: BERT-based deep embedded clustering for topic modeling. In: 2024 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 331–336. IEEE (2024)
9. Cataldo, T., et al.: Researching students' information choices (RSIC): determining identity and judging credibility in digital spaces (2023)
10. Chew, R., Bollenbacher, J., Wenger, M., Speer, J., Kim, A.: LLM-assisted content analysis: using large language models to support deductive coding. arXiv preprint arXiv:2306.14924 (2023)
11. Dai, S.C., Xiong, A., Ku, L.W.: LLM-in-the-loop: leveraging large language model for thematic analysis. arXiv preprint arXiv:2310.15100 (2023)
12. De Paoli, S.: Performing an inductive thematic analysis of semi-structured interviews with a large language model: an exploration and provocation on the limits of the approach. Soc. Sci. Comput. Rev. 08944393231220483 (2023)
13. DeFranco, J.F., Laplante, P.A.: A content analysis process for qualitative software engineering research. Innov. Syst. Softw. Eng. 129–141 (2017). https://doi.org/10.1007/s11334-017-0287-0
14. Dieng, A.B., Ruiz, F.J., Blei, D.M.: Topic modeling in embedding spaces. Trans. Assoc. Comput. Linguist. **8**, 439–453 (2020)

15. Dittrich, Y., John, M., Singer, J., Tessem, B.: Editorial for the special issue on qualitative software engineering research. Inf. Softw. Technol. **49**(6), 531–539 (2007)
16. Easterbrook, S., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. In: Guide to Advanced Empirical Software Engineering, pp. 285–311 (2008)
17. Gamieldien, Y., Case, J.M., Katz, A.: Advancing qualitative analysis: an exploration of the potential of generative AI and NLP in thematic coding. Available at SSRN 4487768 (2023)
18. Gauthier, R.P., Wallace, J.R.: The computational thematic analysis toolkit. In: Proceedings of the ACM on Human-Computer Interaction, vol. 6, no. GROUP, pp. 1–15 (2022)
19. Ghahremanlou, L., et al.: Automating thematic analysis: how LLMs analyze controversial topics. Microsoft J. Appl. Sci. **21**, 69–87 (2024)
20. Glaser, B., Strauss, A.: Discovery of Grounded Theory: Strategies for Qualitative Research. Routledge (2017)
21. Hoda, R.: Socio-technical grounded theory for software engineering. IEEE Trans. Softw. Eng. **48**(10), 3808–3832 (2021)
22. Hoda, R.: Qualitative Research with Socio-Technical Grounded Theory. Springer, Cham (2024)
23. Jahanbakhsh Gudakahriz, S., Eftekhari Moghadam, A.M., Mahmoudi, F.: Opinion texts clustering using manifold learning based on sentiment and semantics analysis. Sci. Program. **2021**(1), 7842631 (2021)
24. Johnson, D.R., Green, A.E., van Hell, J.G., Beaty, R.E.: Creativity in context: thematic profile analysis reveals the explanatory power of themes and culture in creative ideas (2024)
25. Katz, A., Fleming, G.C., Main, J.: Thematic analysis with open-source generative AI and machine learning: a new method for inductive qualitative codebook development. arXiv preprint arXiv:2410.03721 (2024)
26. Khan, A.H., et al.: Automating thematic analysis: how LLMs analyse controversial topics. arXiv preprint arXiv:2405.06919 (2024)
27. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 530–539 (2014)
28. Lenberg, P., Feldt, R., Gren, L., Wallgren Tengberg, L.G., Tidefors, I., Graziotin, D.: Qualitative software engineering research: reflections and guidelines. J. Softw.: Evol. Process **36**(6), e2607 (2024)
29. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11) (2008)
30. Mathis, W.S., Zhao, S., Pratt, N., Weleff, J., De Paoli, S.: Inductive thematic analysis of healthcare qualitative interviews using open-source large language models: how does it compare to traditional methods? Comput. Methods Programs Biomed. **255**, 108356 (2024)
31. McInnes, L., Healy, J., Astels, S., et al.: HDBSCAN: hierarchical density based clustering. J. Open Sour. Softw. **2**(11), 205 (2017)
32. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
33. Narayan, A., Berger, B., Cho, H.: Density-preserving data visualization unveils dynamic patterns of single-cell transcriptomic variability. bioRxiv, pp. 2020–05 (2020)

34. Parker, M.J., Anderson, C., Stone, C., Oh, Y.: A large language model approach to educational survey feedback analysis. Int. J. Artif. Intell. Educ. 1–38 (2024)
35. Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K.: The development and psychometric properties of LIWC2015. Technical report (2015)
36. Rahman, T., Nwokeji, J., Matovu, R., Frezza, S., Sugnanam, H., Pisolkar, A.: Analyzing competences in software testing: combining thematic analysis with natural language processing (NLP). In: 2021 IEEE Frontiers in Education Conference (FIE), pp. 1–9. IEEE (2021)
37. Raza, M.Z., et al.: LLM-TA: an LLM-enhanced thematic analysis pipeline for transcripts from parents of children with congenital heart disease. arXiv preprint arXiv:2502.01620 (2025)
38. Reddy, G.T., et al.: Analysis of dimensionality reduction techniques on big data. IEEE Access **8**, 54776–54788 (2020)
39. Renz, S.M., Carrington, J.M., Badger, T.A.: Two strategies for qualitative content analysis: an intramethod approach to triangulation. Qual. Health Res. **28**(5), 824–831 (2018)
40. Roberts, K., Dowell, A., Nie, J.B.: Attempting rigour and replicability in thematic analysis of qualitative research data; a case study of codebook development. BMC Med. Res. Methodol. **19**(1), 1–8 (2019)
41. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
42. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**, 131–164 (2009)
43. Runeson, P., Host, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering: Guidelines and Examples. Wiley (2012)
44. Ryan, G.W., Bernard, H.R.: Data management and analysis methods. In: Denzin, N.D., Lincoln, Y.S. (eds.) Handbook of Qualitative Research, 2nd edn., pp. 769–803 (2000)
45. Sabbaghan, S.: Exploring the synergy of human and AI-driven approaches in thematic analysis for qualitative educational research. J. Appl. Learn. Teach. **7**(2) (2024)
46. Saldaña, J.: The coding manual for qualitative researchers (2021)
47. Seaman, C., Hoda, R., Feldt, R.: Qualitative research methods in software engineering: past, present, and future. IEEE Trans. Softw. Eng. (2025)
48. Seaman, C.B.: Qualitative methods in empirical studies of software engineering. IEEE Trans. Softw. Eng. **25**(4), 557–572 (1999)
49. Sharp, H., Dittrich, Y., De Souza, C.R.: The role of ethnographic studies in empirical software engineering. IEEE Trans. Softw. Eng. **42**(8), 786–804 (2016)
50. Singh, S.H., Jiang, K., Bhasin, K., Sabharwal, A., Moukaddam, N., Patel, A.B.: Racer: an LLM-powered methodology for scalable analysis of semi-structured mental health interviews. arXiv preprint arXiv:2402.02656 (2024)
51. Xiao, Z., Yuan, X., Liao, Q.V., Abdelghani, R., Oudeyer, P.Y.: Supporting qualitative analysis with large language models: combining codebook with GPT-3 for deductive coding. In: Companion Proceedings of the 28th International Conference on Intelligent User Interfaces, pp. 75–78 (2023)