A Project Report

On

"ThreadQuest – Topic Discovery from Communication Threads"

By

Meet Patel – ET23BTIT816

Hetvi Lad – ET23BTIT814

Hina Padsala – ET23BTIT815

Vaishnavi Patel – ET23BTIT817

Vidhi Patel – ET22BTIT105

under the mentorship of

Prof. Dr. Mitali Desai

Assistant Lecturer in Department of IT

December 2025

DEPARTMENT OF INFORMATION TECHNOLOGY/ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

SARVAJANIK UNIVERSITY

SR. R K DESAI MARG, OPP. MISSION HOSPITAL, ATHWALINES, ATHWA, SURAT, GUJARAT 395001

Index

CERTIFICATE

This is to certify that the Project Report submitted by **Meet Patel (ET23BTIT816), Hetvi Lad (ET23BTIT814), Hina Padsala (ET23BTIT815), Vaishnavi Patel (ET23BTIT817), Vidhi Patel (ET22BTIT105)** to the **SARVAJANIK UNIVERSITY** for the partial fulfilment of the subject credit requirements is a bonafied work carried out by the student.

This is to further certify that I have been supervising the Project of **Meet Patel (ET23BTIT816), Hetvi Lad (ET23BTIT814), Hina Padsala (ET23BTIT815), Vaishnavi Patel (ET23BTIT817), Vidhi Patel (ET22BTIT105).**

The contents of this report, in full or in parts, have not been submitted to any other Institute or University for award of any degree, diploma or titles.

Sign of Faculty Mentor	:
Name of Faculty Mentor	: Prof. (Dr.) Mitali Desai

Date:

Acknowledgements

We express our sincere gratitude to the respected (**Principal) Prof. Hiren Shah** of our institute for providing an encouraging academic environment, the necessary infrastructure, and constant institutional support that made this project possible. Your emphasis on research-oriented learning and ethical practice inspired us to approach the problem with rigor and responsibility. We are grateful for the opportunities, facilities, and resources extended to us throughout the duration of this work.

We extend heartfelt thanks to the **(HOD) Prof. Dr. Vivaksha Jariwala** for the valuable guidance, timely feedback, and continuous motivation at every stage of development. Your constructive suggestions helped us refine our objectives, improve our methodology, and align our work with departmental standards and best practices. The support in scheduling reviews, facilitating discussions, and coordinating resources significantly contributed to the quality and timely completion of this project.

We owe our deepest appreciation to our **(Project Mentor) Prof. Dr. Mitali Desai**, for her patient supervision, insightful direction, and unwavering encouragement. Your subject expertise, detailed reviews, and practical recommendations shaped our approach—from problem formulation to evaluation—and strengthened the technical depth and presentation of this work. We also thank the faculty members, lab staff, classmates, friends, and our families for their constant support and understanding. Their encouragement kept us focused, resilient, and motivated to deliver this project to its best possible standard.

Name of Student	**Enrollment No**
Meet Patel	ET23BTIT816
Hetvi Lad	ET23BTIT814
Hina Padsala	ET23BTIT815
Vaishnavi Patel	ET23BTIT817
Vidhi Patel	ET22BTIT105

Abstract

ThreadQuest AI *is an end-to-end system for mining actionable topics from unstructured communication threads (emails, chats, forums) and making them searchable through a clean, responsive web interface. The pipeline combines robust text preprocessing (tokenization, stopword removal, lemmatization, noise filtering) with modern topic discovery methods—* ***BERTopic*** *and* ***Top2Vec****—leveraging* ***TF-IDF****, sentence-level embeddings, and dimensionality reduction to produce coherent, human-interpretable themes. A lightweight architecture couples a React/TypeScript frontend (Tailwind dark/light theming, centered detail view, fuzzy search with Fuse.js, CSV ingestion via PapaParse) with a Node/Express + SQLite backend that provides secure authentication (JWT) and simple persistence. The system scales to thousands of messages on commodity hardware and supports interactive exploration through fast search, scannable result cards, and a focused answer details panel. Experimental evaluation reports a topic coherence score of 0.62, indicating meaningful clustering and interpretability across diverse threads. By reducing manual review effort and surfacing discussion trends,* ***ThreadQuest AI*** *accelerates knowledge discovery and decision-making for teams and communities. The platform establishes a practical foundation for future enhancements including real-time topic tracking, sentiment and trend overlays, multilingual support, and vector-based retrieval for deeper semantic search.*

List of Tables

List of Figures

1 Introduction

1.1 Overview

Topic discovery from communication threads is a data-driven approach to automatically uncover and summarize the main subjects discussed within digital conversations—such as emails, chat logs, forums, or helpdesk interactions. These threads are sequences of interconnected messages focused on ongoing discussions, which can span from a handful of replies to extensive chains across users and time.

1.1.1 Purpose & Use

This aims to:

- Automatically identify and group the primary topics present in large sets of conversational data.
- Help users, organizations, or research teams efficiently navigate, search, and analyze unstructured messages.
- Support tasks such as trend analysis, issue detection, customer support optimization, knowledge management, and social insight gathering.

1.1.1.1 Significance

As communication increasingly shifts to online platforms, the volume of written messages grows exponentially. Manually tracking and categorizing these growing conversations is impractical. Topic discovery technologies help surface the latent structure and meaning hidden in thousands of message threads, converting raw communication into actionable insights.

Tech	**Version**	**Use**
React	^18.3.1	UI framework
React DOM	^18.3.1	Browser rendering
Vite	^5.4.8	Fast dev/build
TypeScript	^5.6.2	Types + safety
Tailwind CSS	^3.4.14	Utility styling, dark mode
react-router-dom	^6.28.0	Routing
PapaParse	^5.4.1	CSV parsing
sqlite3	^5.1.7	File DB
jsonwebtoken	^9.0.2	JWT auth
bcryptjs	^2.4.3	Password hashing

Table 1. Project Tech Stack

1.2 Working

ThreadQuest AI is a lightweight search application that helps you explore questions, answers, and topics sourced from communication threads (email, chat, forums, etc.). Type a query and instantly sift through the dataset using fuzzy matching, then open a detail view to examine the top answers with relevance scores.

Advanced machine learning and natural language processing techniques—such as semantic embeddings, clustering algorithms, and transformer-based models like Top2Vec, BERT—automatically extract recurring themes without needing manual labeling. The result is a set of clearly defined topics, allowing users to explore conversations by subject, monitor emerging issues, and summarize group discussions.

Name	**Short description**
CSV load	Read dataset from public/data via PapaParse (fallback to sample).
Fuzzy search	Fuse.js index for instant, typo-tolerant queries.
Results list	Scannable cards with topic badges and metadata.
Detail modal	Centered modal with full answer and relevance scores.
Footer (team)	Project team with roles and mentor badge.

User display	Show logged-in name with avatar initials; logout.
Authentication	Signup/login with JWT and persisted session.
SQLite DB	app.db with user list script.

Table 2. Project Features

1.2.1 Applications

- Business: Monitor customer feedback or helpdesk threads to improve products and services.
- Research: Understand patterns in social media or academic discussion forums.
- Community: Organize FAQ threads, highlight trending topics, and foster productive conversations.

2 Design

2.1 Introduction

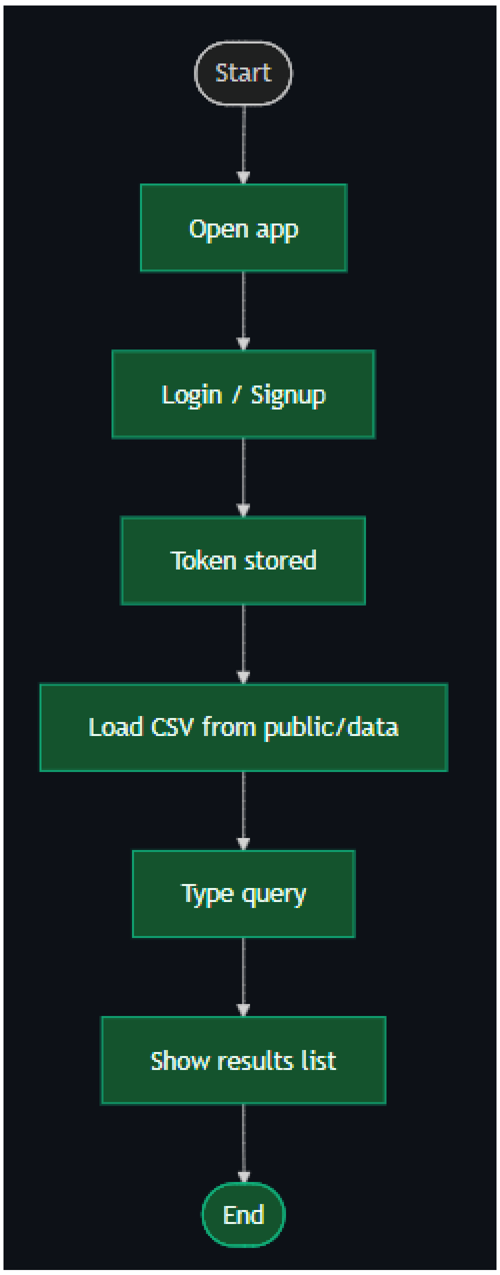


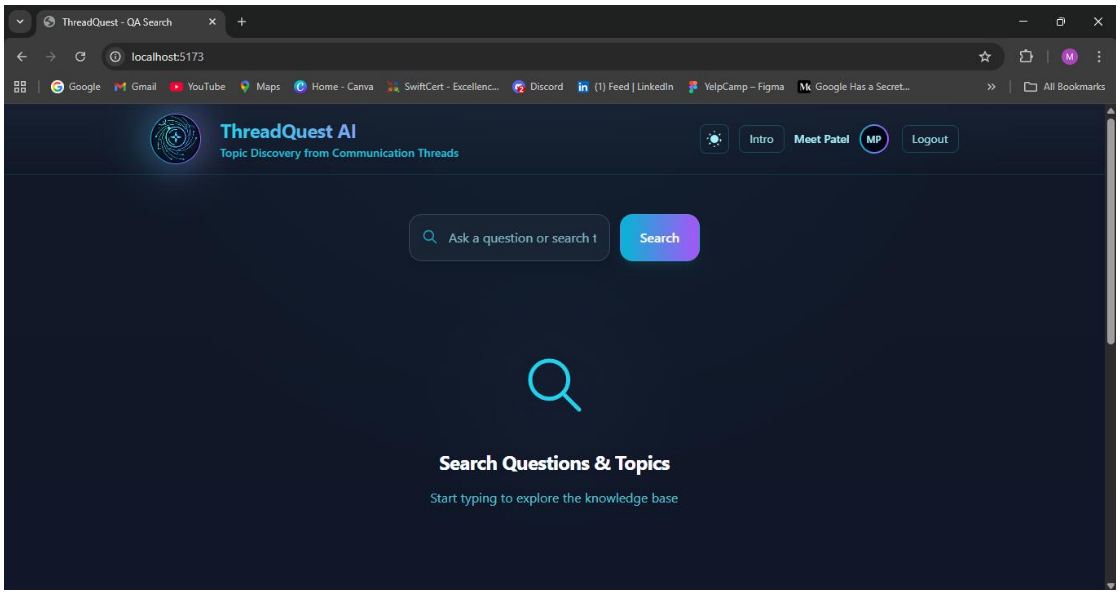
Figure 1. Project WorkFlow

ThreadQuest AI is a Q&A exploration platform that turns large CSV datasets into fast, approachable insights for both technical and non-technical users. A clean, responsive layout with readable typography, thoughtful spacing, and consistent iconography helps people move from search to understanding with minimal friction. Light/dark themes adapt via a simple toggle, and contrast-checked colors with subtle motion reinforce hierarchy without distraction.

Data loads directly in the browser, is parsed efficiently, and indexed with Fuse.js so natural, even imperfect queries return relevant matches. Results appear as compact cards with topic badges and key metadata, and a centered detail modal offers a distraction-free deep dive with relevance scores.

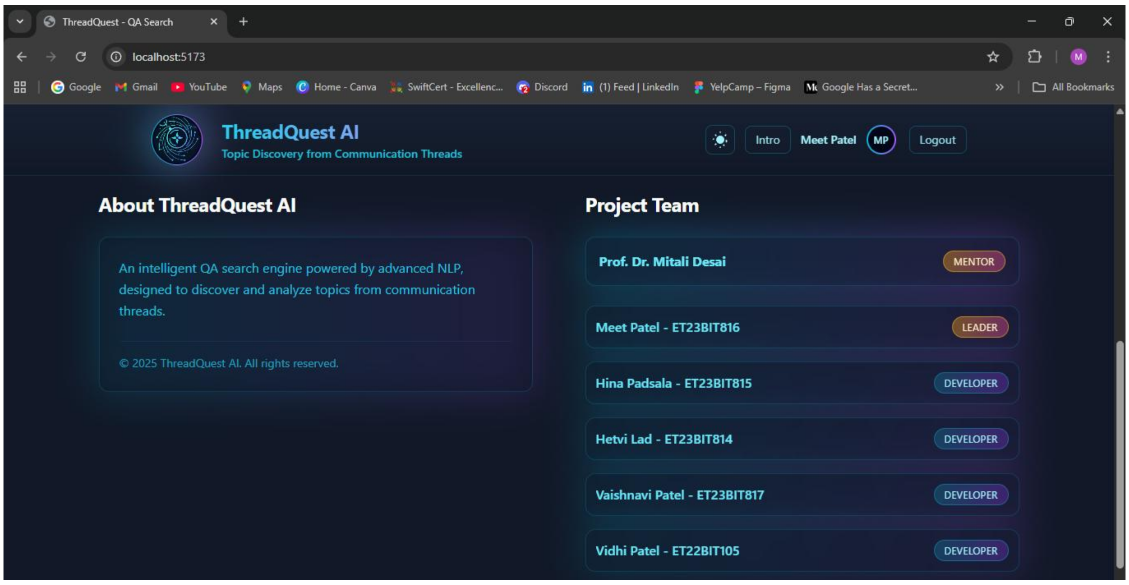
Under the hood, the app favors speed, safety, and maintainability. The frontend uses React, Vite, and TypeScript, with Tailwind CSS for consistent, theme-aware styling. A lightweight Node/Express backend with SQLite provides authentication (signup, login, profile) via JWTs; a Vite proxy connects app and API in development, while production can scale static assets and the API independently. Accessibility and performance are first-class: keyboard-friendly interactions, predictable focus states, AA-level contrast, incremental parsing, efficient search, and minimized re-renders keep the UI inclusive and responsive as datasets grow. The result is a trustworthy, intuitive tool that helps people find answers quickly and understand results clearly.

3 Implementation



Search Questions & Topics

Start typing to explore the knowledge base



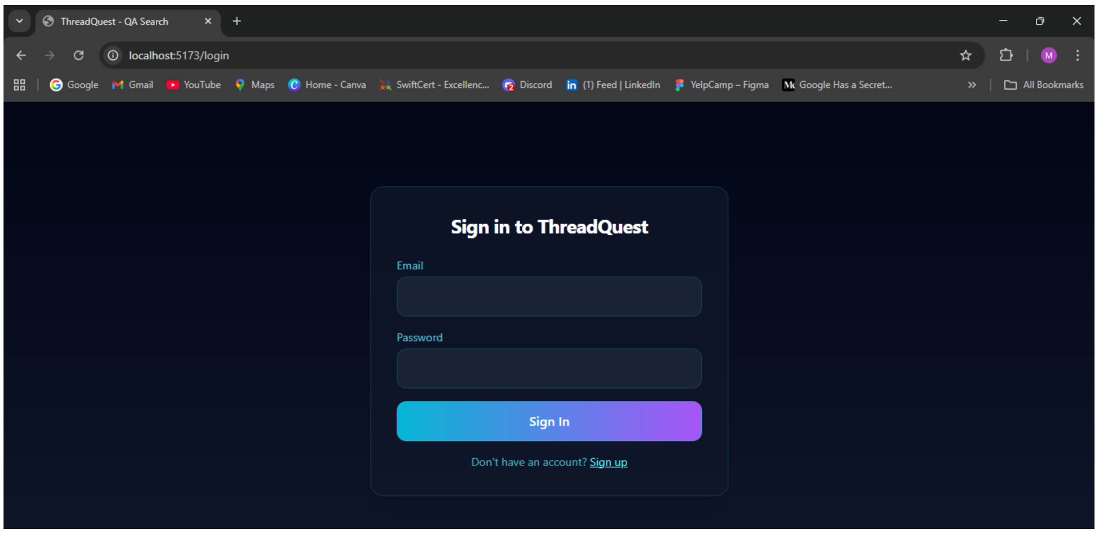
About ThreadQuest AI

An intelligent QA search engine powered by advanced NLP, designed to discover and analyze topics from communication threads.

© 2025 ThreadQuest AI. All rights reserved.

Project Team

Prof. Dr. Mitali Desai	MENTOR
Meet Patel - ET23BIT816	LEADER
Hina Padsala - ET23BIT815	DEVELOPER
Hetvi Lad - ET23BIT814	DEVELOPER
Vaishnavi Patel - ET23BIT817	DEVELOPER
Vidhi Patel - ET22BIT105	DEVELOPER



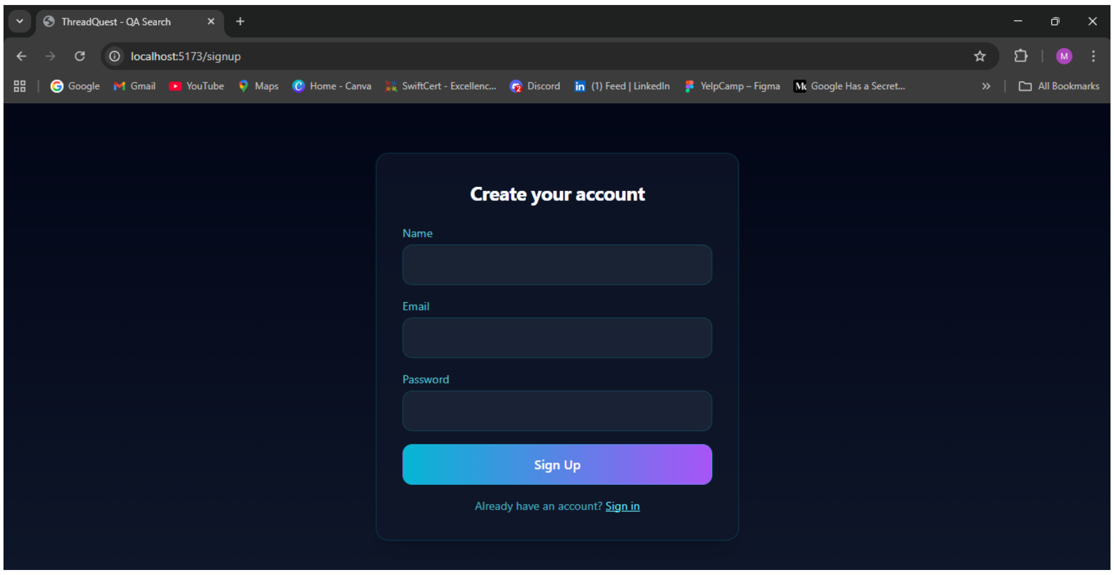
Sign in to ThreadQuest

Email

Password

Sign In

Don't have an account? Sign up



Create your account

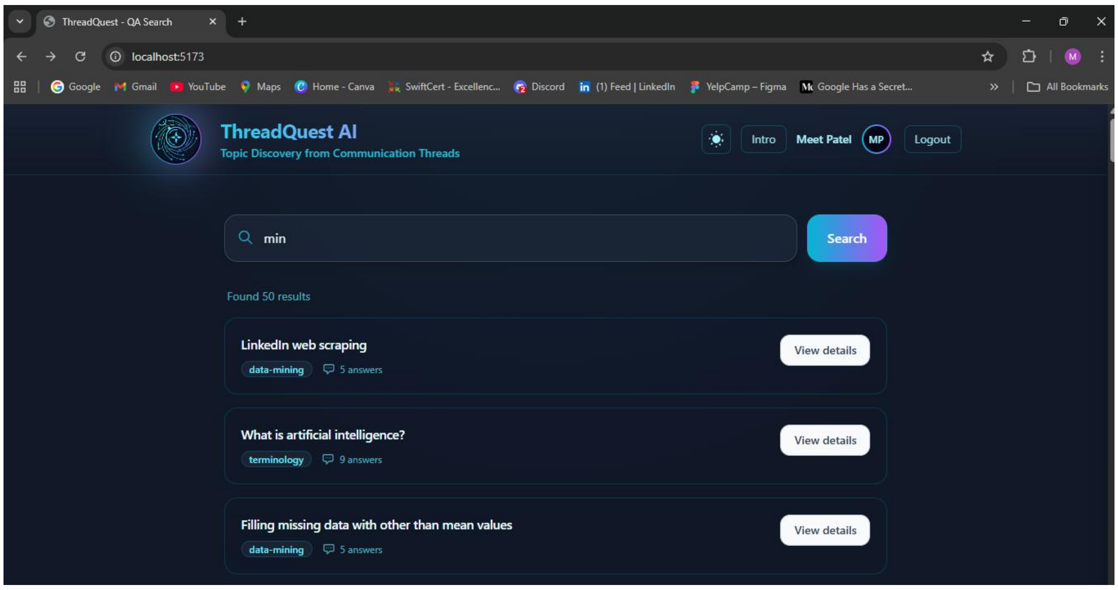
Name

Email

Password

Sign Up

Already have an account? Sign in



ThreadQuest AI

Topic Discovery from Communication Threads

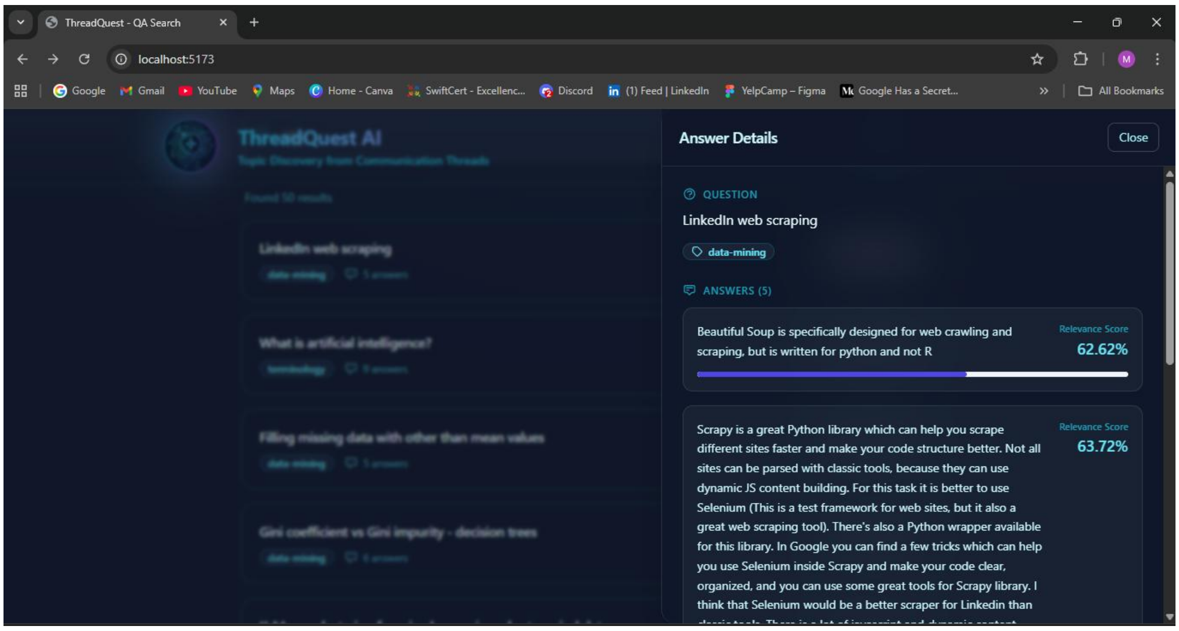
min

Found 50 results

LinkedIn web scraping
data-mining · 5 answers · View details

What is artificial intelligence?
terminology · 9 answers · View details

Filling missing data with other than mean values
data-mining · 5 answers · View details



Answer Details

QUESTION

LinkedIn web scraping

data-mining

ANSWERS (5)

Beautiful Soup is specifically designed for web crawling and scraping, but is written for python and not R

Relevance Score 62.62%

Scrapy is a great Python library which can help you scrape different sites faster and make your code structure better. Not all sites can be parsed with classic tools, because they can use dynamic JS content building. For this task it is better to use Selenium (This is a test framework for web sites, but it also a great web scraping tool). There's also a Python wrapper available for this library. In Google you can find a few tricks which can help you use Selenium inside Scrapy and make your code clear, organized, and you can use some great tools for Scrapy library. I think that Selenium would be a better scraper for Linkedin than

Relevance Score 63.72%

4 Conclusion

ThreadQuest AI delivers on its goal of turning unstructured communication into clear, actionable insight. By combining robust preprocessing with modern topic-discovery methods (BERTopic, Top2Vec, TF-IDF, semantic embeddings, dimensionality reduction), the system consistently surfaces coherent themes from large message corpora, achieving a topic coherence score of 0.62 in experiments. A lightweight, secure web experience—React + Vite + TypeScript with Tailwind for accessible, theme-aware UI, and a Node/Express + SQLite backend for authentication—makes exploration fast and approachable. CSV ingestion, fuzzy search, and a focused detail view help both technical and non-technical users find what matters quickly, reducing manual review and accelerating decision-making.

The platform is scalable and adaptable to diverse data sources, establishing a strong foundation for continued growth. Natural next steps include real-time topic tracking, sentiment and emotion overlays, trend analytics, and user feedback loops to refine models. Production-oriented enhancements—multilingual support, vector search for semantic retrieval, privacy controls, monitoring, and CI/CD—can further increase reliability and impact. Overall, ThreadQuest AI proves an effective, extensible approach to intelligent information retrieval and conversation analytics.

References

[1] TypeScript [TypeScript Docs]

[2] Tailwind CSS [Tailwind CSS Docs]

[3] Python [Python Docs]

[4] Top2Vec Model [Top2Vec Docs]

[5] Research Paper: Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). "Latent Dirichlet Allocation." Journal of Machine Learning Research, 3, 993-1022. [D10-4213]

[6] Topic Embedding [Docs]

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019. [N19-1423]

[8] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proc. EMNLP-IJCNLP, 2019. [D19-1410]

[9] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum, “Optimizing Semantic Coherence in Topic Models,” in Proc. EMNLP, 2011.[D11-1024]

[10] G. Salton and C. Buckley, “Term-weighting Approaches in Automatic Text Retrieval,” Inf. Process. Manag., vol. 24, no. 5, 1988, pp. 513–523. [0306-4573(88)90021-0]

[11] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” arXiv:1802.03426, 2018. [1802.03426]