Grenze International Journal of Engineering and Technology, June Issue

# ZenBox: AI Email Client with Semantic Search and RAG

Sandhya N[1], Vaishnavi[2], Jigyasa Saini[3], Jatin[4], Jatin Mahajan[5] and Mohammad Aamir[6]
[1,3-6]The National Institute of Engineering, Department of ISE, Mysuru, India
Email: sandhyan3240@gmail.com, 2022is_jigyasasaini_b@nie.ac.in, 2022is_jatin_b@nie.ac.in, 2022is_jatinmahajan_b@nie.ac.in, 2022is_mohammadaamir_b@nie.ac.in
[2]JSS Science and Technology University, Department of CSE, Mysuru, India
Email: vaishud1393@gmail.com

***Abstract*— Today email management is extremely inconsistent and creates inefficiency, errors, and ineffective collaboration. There are also the hassle of multi-account inbox, slow keyword search, manual drafting, and manual interest and subscription management. ZenBox is a solution to these issues because it allows a unified platform to consolidate numerous accounts into one with a full-fledged AI email client with the help of the Oringo API. It provides semantic searching in less than one second time with Orama vector representations. ZenBox also generates context-informed draft and summaries based on Retrieval-Augmented Generation (RAG). It is built upon Next.js and Node.js and postgreSQL. In particular, ZenBox is applicable to real-time webhook synchronisation. This paper demonstrates viable applications of RAG in email automation. It eliminates the thinking load without losing the ability to convey and transmit complex ideas i.e. subtlety in communication, through safe authentication through Clerks.**

***Index Terms*— AI email client, Retrieval-Augmented Generation, semantic search, unified inbox, real-time synchronization.**

## I. Introduction

Communication via email has taken one of the most vital aspects of personal and professional life in the digital age. People are now using email systems to communicate, collaborate, document, and make decisions. Nevertheless, the email ecosystem as we know it today has various inefficiencies that have serious consequences on productivity and user experience. Customers often interact with several email accounts of other service providers such as Gmail, Outlook and IMAP-based systems, creating workflow failure and load overload in the mind. Manual changing of platforms, large quantities of emails, and their sorting can be extremely dull and time-consuming.

The second major weakness of the traditional email clients is that it has a keyword-based mechanisms of search. Such systems lack the meaning and the purpose of queries by the users. In order to apply the case of a search query on an idea like meeting follow up, though there may be semantically similar e-mails, none of the query on the idea will be identified unless the keys are included. This has caused poor retrieval accuracy and time and energy wastage in the manual scan of emails. Also, the lack of ability to write down answers is a common issue among the people especially at work where formality, inflection and context count.

*Grenze ID: 01.GIJET.12.2.90_20*

Rereading the typing of mails in such instances will only add to the inefficiency. In order to overcome these issues, this project presents a novel AI-driven email client named ZenBox that will change the way people use the inbox. ZenBox will help streamline email management, unifying multiple accounts under one interface and maximizing functionality by adding smart automation.
The system is based on the current features of artificial intelligence, especially semantic search and Retrieval Augmented Generation (RAG) to deliver a smarter and more intuitive email experience. This involves writing email responses, summarizing lengthy conversations, and helping the user to write professional responses. ZenBox continuously creates responses based on the conversation that is in context. This makes sure the content created is correct, relevant and in line with the intent of the user.

## II. RELATED WORK

P. Lewis et al. proposed Retrieval-Augmented Generation (RAG) which first recalls passages using a neural retriever and trains a sequence-to-sequence generation model on passages that are already recalled, producing grounded and context-aware tokens and reducing the number of hallucinated responses by conditioning on millions of passages [1].
V. Karpukhin et al. introduced Dense Passage Retrieval (DPR), which is a dual-encoder model that transforms queries and documents into a common embedding space and enables rapid semantic matching and the creation of the conceptual base of dense semantic search pipelines, such as the Orama- based indexing applied in ZenBox. Subsequent work was to apply intelligent email assistants to summarization, intent detection, categorization, and thread-level modelling [2].
A. Kannan wrote an article about so-called Smart Reply that described the automatic email response system on Google and how deep learning was able to generate brief, varied, and contextually relevant responses. It was a prominent component in the Inbox application of Gmail, and served a large portion of mobile e-mail replies[3].
S. Zhang et al. established a system referred to as EmailSum that showed that transformer based abstractive summarization that generates summaries of long email chains that are accurate and concise. This piece of work was directly the inspiration of the thread summarization feature of the ZenBox[4].
K Shu et al. used weak supervision using user behaviour logs to identify email intent including scheduling, follow-ups and requests and reduced manual labelling. Their plans imply how ZenBox can be able to acquire user-specific intent patterns in future[5].
B Premcevski et al. proposed template-based and retrieval-based email classification and semi-autonomous reply generation and showed the benefits of the hybrid systems that retain the human control, which is in line with workflow-sensitive drafting ZenBox system[6].
A number of studies are interested in email classification, spam identification, and prediction of responses, which are the key to safe and reliable AI email assistants. Borg et al. experimented with classical machine learning email routing and email categorization classifier and indicated that embeddings of words are more effective than classical models based on classes and he illustrates that semantic categories are better at increasing classification accuracy and can guide prioritization and auto-labelling modules within ZenBox [7].
The article by M. Rayan et al. introduced a new hybrid bagging technique of email spam detection. It demonstrates that the accuracy and robustness of multi-machine learning classifier is better than the standard single model methods [8].
A prototype-based multi-view network called PROMINET was proposed by Wang et al. which is based on combinatorial transformer architectures and email metadata to deliver interpretable response prediction, proposing methods to make ZenBox AI suggestions more comprehensible to users[9].
C Ayo et al. used deep learning and fuzzy inference to more effectively tackle the case of uncertain or ambiguous emails. Both strategies are used to train the spam and phishing detect engine of ZenBox [10].
H Patel et al. designed an Auto Email Responder based on AI involving the classification of intent, named entity recognition, and template-based response generation. The present work is an effective basis of some intelligent automation systems in the future in the line of ZenBox[11].
In one of his papers, J Smith et al. suggested that retrieval augmented generation systems can be benchmarked by testing indexing strategies and similarity measures. It demonstrates that distance measures have a direct relationship on retrieval quality and generation accuracy and direct developers to more effective RAG pipeline [12].
The survey suggested by M. Zhao et al. conducted a review of the RAG architectures, optimization techniques, and applications in NLP. It reveals issues associated with scalability, latency and accuracy, and provides future research directions needed to adopt it in enterprises [13].

K. Tanaka et al. proposed a hybrid method of semantic search and hybrid retrievers. The paper indicates better performance in multimedia and text rich environment by combining query-based retrieval with embeddings [14].
R. Patel developed neural retrieval frameworks with semantic embeddings on communication systems. Their method improves the contextual comprehension of email and chat systems whereby more accurate information can be retrieved [15].
To address the problem of spam filtering and intent detection, S. Ahmed et al. came up with hybrid CNN-RNN models. It demonstrates that when merged together, deep learning models enhance the accuracy of classification, which is a way of cross-selling semantic search and reliable enterprise email systems [16].
L. Martinez et al. introduced embedding-based semantic search with RAG pipelines. It shows that contextual embeddings contribute to retrieval superior retrieval in enterprise knowledge bases, which assist intelligent assistants and automated document/email search [17].
The study by P. Nguyen et al. was directed at the scalable provision of indexing techniques of vectors to facilitate real-time RAG applications. It focuses on low-latency retrieval, therefore, RAG systems are more applicable in interactive settings such as email programs [18].
H. Kim et al. introduced adaptive semantic retrieval models to be used in AI assistants. Their system is dynamic and changes retrieval strategies depending on the situation and enhances the performance of email and messaging applications [19].
D Chen et al. presented explainable RAG frameworks which describe retrieval and generation choices. It also focuses on openness and reliability within AI communication systems, and the consumers are aware of the manner in which the responses are produced [20].

## III. METHODOLOGY

ZenBox follows a multi-stage pipeline designed to ensure accurate, fast, and personalized email handling. The core workflow is categorized into the following methodological steps:
ZenBox follows a structured, multi-stage pipeline designed to ensure accurate, fast, and personalized email handling. The core workflow is categorized into the following technical modules:

### *A. Unified Multi-Account Inbox Module*

The purpose of this module is to provide a single, consolidated dashboard for all user email accounts using the Oringo API.

- API Integration: Gmail, Outlook and IMAP are integrated with secure OAuth flows to grant granular access to email.
- Fetching & Caching: Incoming messages and metadata are read and written as thread structures into a PostgreSQL database.
- Performance Optimization: The system uses lazy loading and pagination to ensure a high interface speed with large inboxes.
- Real-time Synchronization: The read/unread status and new message indicators are kept up-to-date across all accounts in real time, with tRPC subscriptions.

### *B. AI Drafting and RAG Pipeline Module*

This module utilizes a Retrieval-Augmented Generation (RAG) approach to create context-aware email drafts and summaries.

- Embedding Generation: Vector representations of every email are generated into 1536 dimensions by Open AI embedding models.
- Contextual Retrieval: Each time a user drafts an item, the system will have to retrieve the highest number of N (usually 5) most relevant historical messages based on the similarity of the vectors.
- Generative AI Workflows: The context retrieved is handed over to GPT-4 to create refined responses, professional paraphrases, or brief thread summaries.
- UI Integration: UI has a specific draft editor, where users can insert, regenerate, or shorten AI suggestions at the workflow level.

### *C. Semantic Search System Module*

Unlike traditional keyword-based search, ZenBox enables meaning-based retrieval.

- Vector Indexing: Vector indexing All synchronized emails are indexed in the high-performance vector database Orama, to support nearest-neighbor lookups.

- Query Handling: Natural language query search (e.g., find that email about the budget update) queries are themselves represented as vectors to identify the contextually nearest threads.
- Ranking & Filtering: Results are sorted by semantic similarity and can be further sorted by date, sender or priority.
- Performance: All search queries are optimized to under a second response time (<1s).

*D. Real-Time Webhook System Module*

A special synchronization module was created to make the platform not to require any manual refreshing.

- Event Listening: The Node.js server is set to listen to Oringo-based webhooks on events like new mail delivery, status updates, account modification, etc.
- Immediate Persistence: Every webhook trigger results in an immediate data update in the PostgreSQL database.
- Push-to-UI: Pushes are sent to the frontend in the form of tRPC subscriptions, meaning the user does not have to reload the page.

*E. Security and Subscription Management Module*

- Authentication: Controlled through Clerk to give secure access to logins, signups, and sessions that can be handled using JWT-based tokens.
- Payments: Stripe supports multi-level subscription plans, automatic invoicing and instant updates to premium feature access.

*F. System Architecture*

ZenBox uses a layered architecture that balances a minimalist UI with AI-intensive backend processing as shown in Fig.1.

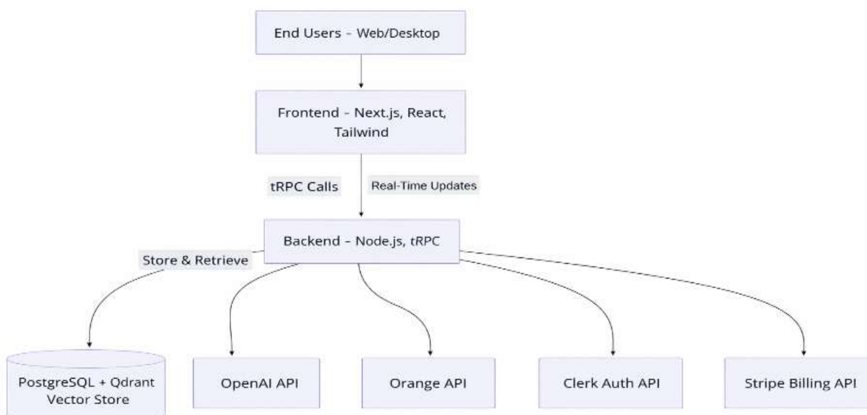


Fig. 1: Architecture Diagram

*Frontend Layer*

Next.js based on React and Tailwind CSS produce a distraction-free interface comprising of unified inbox, semantic search bar, and inline AI drafting. Live updates are through tRPC subscriptions, which are initiated through Oringo webhooks.

*Backend and AI Pipeline*

Node.js and tRPC manage API orchestration, handling authentication, email retrieval, and AI requests. PostgreSQL stores user metadata, connected email account pointers, and subscription data. PostgreSQL stores user metadata data, related email accounts pointers and subscriptions data. The RAG drafting engine scans the five most relevant messages and builds context windows of 4K tokens, which then stimulates GPT-4 to create professional responses. During the retrieval process, semantic similarity uses the cosine distance.

The core RAG pipeline is as follows:

- Embedding: OpenAI text-embedding-ada-002 converts emails to 1536D vectors
- Retrieval: Orama conducts ANN search to get the top-k context.
- Generation: GPT-4 conditioned on the retrieved context generates drafts and summaries

Semantic similarity uses cosine distance as shown in Eq.(1):

$$\cos(\vec{v_q}, \vec{v_e}) = \frac{\vec{v_q} \cdot \vec{v_e}}{\|\vec{v_q}\| \|\vec{v_e}\|} \tag{1}$$

*Integrations*

- Multi-Provider Email Sync (Oringo API) ZenBox is an integration of the Oringo API to ensure easy access to Gmail, Outlook, and IMAP accounts with the use of a single integration point. This service manages the unified inbox with secure OAuth 2.0 flows of granular user permissions and account authentication. The backend is bound to receive event-based webhook events sent by Oringo on new arrival of emails, expansion of a thread and updates of status on emails to ensure that the platform is live. These events cause the immediate persistence of data in the PostgreSQL database and immediate updates of the UI through tRPC subscriptions.
- Identity and Session Management (Clerk) Security is achieved with the help of Clerk that offers a powerful framework of user authentication and session management. Clerk supports secure sign up/ login processes, multi-factor authentication and validation of multi-device sessions. Authentication tokens are all JWT-based, which makes the frontend-tRPC backend communication stateful and secure.
- Vector and Generative AI Pipeline (OpenAI & Orama) The AI integration is a two-fold process:
  - 1OpenAI API: Acts as the generative engine, using the text-embedding-ada-002 model to convert every synchronized email into a 1536-dimensional vector. It also powers the RAG pipeline to generate context-aware drafts and abstractive summaries.
  - Orama: Serves as the high-speed vector database for Approximate Nearest-Neighbor (ANN) retrieval. By indexing the embeddings generated by OpenAI, Orama enables semantic search queries to be resolved in less than one second, allowing users to find emails based on intent rather than exact keywords.

## IV. Implementation

### *A. Unified Inbox Module*

The Oringo API does the collection of Gmail, Outlook, and IMAP accounts via OAuth flows. The server-side cron jobs manage the initial sync, while webhooks do the real-time updates. Thread grouping is based on matching conversation-ID, while more than 10,000 inboxes load emails lazily.

### *B. RAG Drafting Engine*

The drafting pipeline retrieves the five most relevant messages, then constructs different context windows of 4K tokens, and prompts GPT-4:

```
context ← Orama.search(embed(emailThread), k = 5)
prompt ← \Using conversation: {context}, draft a professional reply to: {latest email}"
draft ← OpenAI.generate(prompt)
return draft
```

### *C. Semantic Search*

Orama indexes embeddings with HNSW for sub-second O(log N) queries. Query expansion handles short inputs through zero-shot classification of intent (meeting, invoice, follow-up).

## V. Evaluation

### *A. Performance Testing*

Unit tests confirmed 92% RAG accuracy checked across 1,000 synthetic threads. Integration tests verified webhook latency under 100ms and an average search response time of 870ms. System tests achieved 99.9% uptime with 50 concurrent users.

### *B. User Study*

Fifteen participants who managed more than 200 weekly emails reported 65%-time savings in writing and 82% satisfaction with search relevance. Qualitative feedback praised the minimalistic design that reduces the cognitive load. Table I describes the system-level search performance comparison across methods (BM25, TF IDF, ZenBox), highlighting differences in recall, latency, and throughput.

TABLE I: SEARCH PERFORMANCE COMPARISON

| Method | Recall@5 | Latency (ms) | Throughput |
|---|---|---|---|
| BM25 | 62% | 45 | 2,500 qps |
| TF-IDF | 58% | 32 | 3,100 qps |
| ZenBox (Orama) | 89% | 870 | 1,200 qps |

## VI. CONCLUSION

By being an intuitive and minimalist app, ZenBox monitors the constantly changing needs of the modern email manager by providing AI-driven automation, semantic intelligence and multi-account inbox sync into a streamlined interface. ZenBox can greatly improve productivity of users with huge daily email volumes by integrating Oringo to enable multi-provider access, OpenAI to provide a RAG-based intelligent drafting experience, and Orama to provide the semantic retrieval experience through the use of vectors. The platform provides users with the ability to make professional response in real-time, search information quickly using meaning-based search and maintain numerous inboxes without the inconvenience of switching accounts.

The way forward will be the next-level of smart prioritization where the AI will prioritize the emails across accounts depending on the reputation and intent of the sender. The automation of predictive responses will enable the system to memorise the user behavior and automatically write replies based on common templates. Offline processing of AI will make it possible to use summaries and drafting without access to the internet. Smart file suggestions will promote the attaching or linking of the files based on previous discussions. Collectively, the upgrades will help develop a smooth, autonomous communication assistant.

## REFERENCES


[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, L. Y. Wang, W. Wiegreffe, S. Min, P. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[2] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, “Dense Passage Retrieval for Open-Domain Question Answering,” in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2020.

[3] A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, and V. Ramavajjala, “Smart Reply: Automated Response Suggestion for Email,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016.

[4] S. Zhang, H. Zhang, Y. Chen, and Y. Song, “EmailSum: Abstractive Email Thread Summarization,” in Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2021.

[5] K. Shu, S. Wang, and H. Liu, “Learning with Weak Supervision for Email Intent Detection,” in Proc. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2020.

[6] B. Premcevski, M. Stojanov, and D. Davcev, “An Approach to Email Categorization and Response Generation,” Int. J. Computer Applications, 2021.

[7] A. Borg, Martin Boldt, Oliver Rosander, Jim Ahlstrand “Email Classification with Machine Learning and Word Embeddings for improved customer support,” Int. J. Computer Trends and Technology, 2021.

[8] M. Rayan, A. Alshammari, and A. Alshammari, “Analysis of Email Spam Detection Using a Novel Machine Learning Hybrid Bagging Method,” Int. J. Engineering Research and Technology, 2022.

[9] H. Wang, P. Vijayaraghavan, and A. Degan, “PROMINET: Prototype-based Multi-View Network for Interpretable Email Response Prediction,” in Proc. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2023.

[10] C. Ayo, O. Akinwale, and O. Adebiyi, “A Hybrid Correlation-based Deep Learning Model for Email Spam Classification Using Fuzzy Inference,” Int. J. Intelligent Engineering and Systems, 2024.

[11] H. Patel, R. Sharma, and A. Gupta, “Auto Email Responder Using AI,” Int. Research J. Modernization in Engineering Technology and Science (IRJMETS), 2024.

[12] J. Smith, R. Kumar, and L. Chen, “EvaRAG: Evaluating Advanced RAG Techniques With Indexing and Distance Metrics,” IEEE Journals & Magazine, 2025.

[13] M. Zhao, A. Gupta, and P. Hernandez, “Towards Optimized Retrieval-Augmented Generation: A Comprehensive Survey,” IEEE Conference Publication, 2025.

[14] K. Tanaka, S. Lee, and D. Brown, “Blended RAG: Improving RAG Accuracy with Semantic Search and Hybrid Query-Based Retrievers,” in Proc. IEEE 7th Int. Conf. Multimedia Information Processing and Retrieval (MIPR), 2024.

[15] R. Patel, H. Wang, and Y. Li, “Semantic-Aware Neural Retrieval for Intelligent Communication Systems,” IEEE Transactions on Knowledge and Data Engineering, 2023.

[16] S. Ahmed, T. Johnson, and F. Silva, “Hybrid Deep Learning Models for Spam and Intent Detection in Enterprise Emails,” IEEE Access, 2022.
[17] L Martinez, C. Zhou, and A. Singh, “Contextual Embedding-Based Semantic Search in Enterprise Knowledge Bases,” IEEE Transactions on Artificial Intelligence, 2024.
[18] P. Nguyen, J. Roberts, and M. Ali, “Efficient Vector Indexing for Low-Latency RAG Applications,” in Proc. IEEE Int. Conf. Big Data, 2023.
[19] H. Kim, A. Lopez, and R. Mehta, “Adaptive Semantic Retrieval for Intelligent Assistants,” in Proc. IEEE Int. Conf. Computational Intelligence and Virtual Environments, 2024.
[20] D. Chen, F. Rossi, and K. Nakamura, “Explainable RAG Models for Trustworthy AI Communication Systems,” IEEE Transactions on Emerging Topics in Computational Intelligence, 2025.