IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 6, December 2025, pp. 4600~4613
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i6.pp4600-4613

Semantic search-enhanced healthcare chatbot for hospital information management system using vector database and transformer models

Erda Guslinar Perdana[1,2], Arya Adhi Nugraha[3]
[1]Software Engineering Applications Study Program, School of Applied Science, Telkom University, Bandung, Indonesia
[2]Center of Excellence for Inspiring Digital Transformation for Social Innovation (InsPiRo), Research Institute of Sustainable Society, Telkom University, Bandung, Indonesia
[3]Product Development and Technology Division, Medxa, Bandung, Indonesia

Article Info

Article history:

Received Mar 5, 2025
Revised Oct 21, 2025
Accepted Nov 8, 2025

Keywords:

Healthcare chatbot
Intent recognition
Multilingual embeddings
Semantic search
Vector database

ABSTRACT

Healthcare chatbots are increasingly used to assist hospital staff, yet most existing systems rely on rule-based or generic machine learning (ML) approaches that lack the ability to comprehend natural language queries, while proprietary deep learning systems often incur high licensing costs. This work addresses this gap by proposing a cost-effective and scalable semantic vector retrieval solution for user intent recognition in a hospital information management system (HIMS) helpdesk chatbot. The MPNet-based transformer model is employed to convert user inquiries and predefined intents into feature vectors, enabling highly accurate natural language understanding through cosine similarity retrieval within a dedicated vector database. The proposed vector search method was validated via an ablation study, achieving an accuracy of 0.70 for intent recognition, which demonstrates a significant performance gain of 28.0 percentage points over a traditional keyword-based search baseline. Usability testing across developer and doctor groups yielded an average score of 7.78 on a 10-point Likert scale. This study concludes that integrating semantic vector retrieval with a vector database is highly effective for recognizing specialized clinical intents, offering a more accurate solution that significantly reduces the manual helpdesk workload and enhances 24-hour assistance in healthcare.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Erda Guslinar Perdana
Software Engineering Applications Study Program, School of Applied Science, Telkom University
Bandung, West Java Province, Indonesia
Email: erda@telkomuniversity.ac.id

1. INTRODUCTION

The hospital information management system (HIMS) is a multifaceted information system due to its engagement with several user groups from both internal and external hospital environments. The large number of user groups drives the need for a capable helpdesk system to serve the needs of HIMS users optimally. Chatbots may be integrated into the HIMS helpdesk system, providing automation and 24-hour assistance, alongside cost reduction and the capacity to manage a substantial volume of customers [1]–[3].

Chatbots may typically be constructed using rule-based or machine learning (ML) approaches [4]. Rule-based chatbots are the most prevalent sort of chatbots utilized in the corporate sector nowadays [5]. Rule-based chatbots are chatbots that are developed with a fixed database of question-and-answer pairs, which results in a significant limitation: their inability to comprehend user inquiries expressed in natural language [5]. Conversely, although ML-based chatbots like ChatGPT offer similar benefits, their potential to generate artificial hallucinations makes them less appropriate for integration into help desk systems [6]. Alkaissi and McFarlane [7] elucidates that artificial hallucinations are the result of machines, such as chatbots, generating sensory experiences that appear genuine but do not correspond to any real-world input. The chatbot's artificial hallucination output is inadequate for helpdesk systems that need precise responses. Commercial rule-based chatbot frameworks such as Google Dialogflow, IBM Watson, and Rasa platform have made strides in addressing the limitations of basic rule-based systems by offering more sophisticated capabilities. However, these solutions typically operate as cloud-based services and often incur licensing costs.

Traditional keyword-based search methods struggle to interpret the nuances of natural language, which limits their usefulness in conversational artificial intelligence (AI). To address this, modern systems adopt semantic search powered by dense retrieval (DR) techniques [8]. In DR, both user queries and intent representations are transformed into high-dimensional vector embeddings using deep learning models.

Most DR systems use bi-encoder architectures [9] such as the MPNet-based model used in this study where the query and intent are encoded separately. This setup is highly scalable, allowing intent vectors to be precomputed and quickly compared with new queries using cosine similarity. For tasks that demand higher accuracy, a two-stage retrieval process is often employed. The first stage uses a fast bi-encoder for initial filtering, followed by a cross-encoder that re-evaluates the top results by jointly modeling the relationship between the query and the intent [10]. Although this approach enhances precision, it also increases computational overhead, which remains a key challenge in ongoing research.

Applying natural language processing (NLP) in healthcare introduces additional challenges, particularly in multilingual contexts where clinicians may switch between languages [11] or use specialized medical terms. To manage this complexity, models based on sentence-bidirectional encoder representations from transformers (BERT) [12], such as MiniLM [13] and MPNet [14] variants are trained to generate multilingual sentence embeddings that align semantic meanings across languages. These embeddings ensure that queries in different languages can be understood and processed consistently. Model performance is typically evaluated using large-scale benchmarks like the massive text embedding benchmark (MTEB) [15], which assesses how well these embeddings generalize across languages and domains.

In production systems, the scalability and speed of vector matching play a crucial role. Performing exhaustive k-nearest neighbor (KNN) searches on large datasets can be computationally expensive. As a result, modern systems employ vector databases such as Milvus, which implement approximate nearest neighbor (ANN) [16] algorithms most notably hierarchical navigable small worlds (HNSW) [17]. These methods slightly compromise accuracy but achieve significant performance gains, making ANN-based vector databases the preferred solution for real-time semantic vector retrieval applications.

This work proposes an alternative approach: a text-based rule-based chatbot leveraging a vector database management system and deep learning-based NLP [18]. This solution distinguishes itself by offering a more cost-effective as it is developed using open-source and a low-code environment (Oracle Apex). The key contributions of this work include:

i) Demonstrating the effectiveness of a multilingual MPNet model in identifying complex bilingual (Indonesian–English) clinical intents,
ii) Measuring a 28-percentage-point improvement in response accuracy when using semantic vector retrieval compared to a lexical baseline (Jaccard similarity) within an HIMS helpdesk setting,
iii) Developing a scalable, fully self-hosted architecture built on Oracle Apex, Django, and Milvus, offering healthcare institutions an affordable alternative to commercial cloud-based natural-language understanding (NLU) systems.

A comparison of different chatbot architectures, detailing their search methodology, indexing framework, performance focus, and cost/scalability, is provided in Table 1.

Table 1. Comparison of chatbot architectures

Chatbot system/ architecture	Search methodology	Indexing/framework	Performance focus	Cost/scalability
Traditional rule-based	Lexical/exact match	SQL/RDBMS	Low accuracy, high speed	Low cost, poor scalability
Basic ML	Intent classification (NLU)	-	High accuracy, medium speed	Cost varies, poor scalability
Modern vector search	Semantic retrieval (Bi-encoder)	Vector databases (e.g., Milvus, FAISS, weaviate, elastic search)	High accuracy, scalable performance	Moderate cost (infrastructure), highly scalable
This study	Semantic retrieval (MPNet)	Milvus database	Milvus vector databases, Oracle Apex	Optimized accuracy for HIMS intents (acc: 0.70)

2. METHOD
2.1. User intent dataset preparation
We performed an analysis of the HIMS user manual to extract information pertinent to doctors inside the HIMS framework. The data was subsequently aggregated into a compilation of 11 user intents as shown in Table 2. For each user intent, 13 realistic user query samples were manually formulated (10 for training, 3 for testing) based on common doctor inquiries, resulting in 143 total records.

Table 2. User intent dataset

No	User intent	Training samples	Testing samples
1	Surgery schedule	10	3
2	Doctor practice hours	10	3
3	Outpatient nurse shift schedule	10	3
4	Doctor visit schedule	10	3
5	Medical service payment detail	10	3
6	Online reservation patient list	10	3
7	Consulted patient information	10	3
8	Polyclinic patient list	10	3
9	Medical procedure details	10	3
10	Doctor feedback on chatbot usage	10	3
11	Chatbot feature overview	10	3

2.2. Embedding models evaluation
Embedding models were utilized to create vector data from user intent and user query to make semantic vector retrieval with cosine similarity being possible. Five embedding models were selected as candidates based on their reported performance on MTEB and their documented multilingual capabilities [15]. The inclusion of multilingual models is crucial given the potential for code-switching or variations in clinical terminology often found in healthcare contexts [11].

To determine the most suitable model for user intent dataset, performance evaluation was needed. The performance evaluation was conducted by training a logistic regression classifier to assess the quality of the generated user intent embeddings [19]. This classification task serves as an objective proxy to determine the embedding model best capable of distinguishing between the 11 user intents in the vector space. The model selected based on precision, recall, and F1-score will then be implemented to facilitate the final semantic vector retrieval. The five embedding models' candidates are: i) all-MiniLM-L6-v2, ii) paraphrase-multilingual-MiniLM-L12-v2, iii) paraphrase-multilingual-mpnet-base-v2, iv) distiluse-base-multilingual-cased-v2, and v) sentence-transformers/LaBSE.

2.3. Semantic vector retrieval vs keyword matching comparison
We conducted an ablation study to quantify the improvement in intent recognition gained by using semantic vector retrieval with cosine similarity over a traditional keyword-based search baseline. For the semantic vector retrieval component, the selected embedding model was used to vectorize all user intent samples. Nearest neighbor classification was then applied, utilizing cosine similarity to find the single k=1 nearest neighbor (the predicted intent class) for each test sample.

For the keyword-based search baseline, the Jaccard similarity algorithm was chosen to measure the token overlap between the user query and the training intent samples. The intent class with the highest Jaccard score was selected as the prediction. Accuracy was calculated for both methods, allowing us to quantify the performance gain of the semantic vector retrieval approach for intent recognition.

2.4. Threshold sensitivity analysis
To determine the optimal cosine similarity threshold for intent matching and to analyze its effect on chatbot performance, a threshold sensitivity analysis was performed. The selected embedding model was utilized to convert intent sample dataset into embedding vector. For each intent sample test dataset, the cosine similarity between the test embedding and its top retrieved candidate was computed using the KNN algorithm with cosine similarity. A series of thresholds ranging from 0.40 to 0.95 (in increments of 0.01) was evaluated. A match was only accepted if the highest similarity score among the k candidates was greater than or equal to the tested threshold (τ). The intent corresponding to the highest score above τ was designated as the prediction; otherwise, the query was classified as "No match found" (rejection). For each threshold value, precision, recall, and F1-score were computed.

2.5. Software development
The agile development methodology was employed to develop the chatbot application. This method is characterized by its incremental nature (small release versions with fast cycles), cooperative nature (developers

and users work together with intense communication), and adaptability [20], [21]. The solution utilizes a low code technology platform to improve the efficiency of software development [22], specifically employing the Oracle Apex development platform for the front-end interface and core business logic integration.

2.6. Chatbot evaluation

Usability evaluation was employed to assess the resulting chatbot application. Usability is defined by ISO 9241-11:1998 as "the degree to which a product can be utilized by specific users to achieve specific objectives with efficiency, effectiveness, and satisfaction in a specific context of use" [23]. According to the definition, three metrics are utilized in the evaluation of this study: effectiveness, efficiency, and satisfaction. Each measure includes three properties that are checked using a questionnaire with a Likert scale ranging from 1 to 10 [24]. These nine attributes refers to a study conducted by Nicole and Morgan [25] in conducting usability testing on chatbots. The questionnaire respondents comprised two groups: six individuals from the HIMS application development team (for technical validation) and fifteen members of the doctor group (for end-user validation).

2.7. Active learning loop

To ensure the system remains accurate and adaptive while utilizing a fixed embedding model, we propose implementing an active learning (AL) loop to continuously refine the user intent knowledge base in the vector database. This mechanism is critical for addressing instances of semantic ambiguity and coverage gaps. The AL loop focuses on identifying high-value queries by monitoring three signals: confidence threshold violation (query is too distant from all current intents); ambiguity sampling (top similarity scores are too close); and user escalation/feedback signal (the query results in immediate negative user feedback). These flagged queries are routed to the HIMS content management interface where an expert assigns the correct intent and adds the query as a new intent sample. Finally, the 'Vectorize all intent' function is executed via the Django API to generate the vector for the new sample and insert it into the Milvus vector database. This process iteratively expands and clarifies the boundaries of the intent clusters in the vector space, enhancing semantic retrieval accuracy over time without the necessity of expensive model retraining.

2.8. System performance evaluation

System-level performance metrics were evaluated to ensure that the semantic vector retrieval can operate efficiently in real-time healthcare chatbot scenarios. Performance evaluation focused on latency and throughput consistency under concurrent query loads. Latency refers to the average time taken for Milvus to retrieve the top-1 semantic match from a vector collection after receiving a user query. Throughput represents the number of successful retrieval operations per second. The benchmarking environment consisted of a Milvus 2.3.21 instance hosted on a cloud server equipped with 16 GB of RAM, 8 vCPUs, and an SSD-based storage backend. Load testing was simulated using a Python-based asynchronous client built with locust and aiohttp, generating concurrent query requests ranging from 1 to 100 parallel users. Each client issued random, pre-embedded queries from the intent sample test dataset, and the average end-to-end latency (from request to retrieval result) was measured over 1,000 queries per concurrency level.

3. RESULTS AND DISCUSSION

3.1. Embedding model evaluation

We trained logistic regression classifier from our user intent training sample which has been generated into vector data by each candidate model. Vectorized user intent testing sample then was used to evaluate the trained classifier. The evaluation result as shown in Table 3 indicates that paraphrase-multilingual-mpnet-base-v2 outperformed another candidate models in all evaluation metric.

Table 3. Embedding model evaluation result

No	Model	Accuracy	F1-score	Precision	Recall
1	all-MiniLM-L6-v2	0.5455	0.5102	0.5795	0.5455
2	paraphrase-multilingual-MiniLM-L12-v	0.6364	0.5686	0.6011	0.6364
3	paraphrase-multilingual-mpnet-base-v2	0.7576	0.7102	0.7212	0.7576
4	distiluse-base-multilingual-cased-v2	0.6061	0.5554	0.5697	0.6061
5	sentence-transformers/LaBSE	0.6364	0.5962	0.6515	0.6364

3.2. Error analysis

Further analysis was conducted for paraphrase-multilingual-mpnet-base-v2 to identify intents misclassifications. We found that 6 of 11 intents were classified with 100% accuracy. The operational and patient list intents surgery schedule, doctor practice hours, outpatient nurse shift schedule, online reservation patient list, polyclinic patient list, and consulted patient information were clearly separable in the embedding space. These intents represent core clinical and administrative queries that the embedding model successfully distinguished.

High-confusion intents, which experienced the majority of the errors, were medical procedure details, medical service payment detail, doctor feedback on chatbot usage, and chatbot feature overview. The misclassification suggests a strong semantic overlap between these intents. The classification results for these problematic intents are detailed in Table 4. Figure 1 shows the complete confusion matrix. Table 5 further illustrates the specific nature of these errors by providing the misclassified test queries and the model's incorrect prediction. The analysis shows that, while the model is robust for the majority of operational tasks, the overlap between specific detail-oriented questions (like procedure steps or medical service fee) and system-meta questions (chatbot features/feedback) requires further attention.

Table 4. Misclassification analysis

True intent	Correctly classified	Misclassified	Errors and analysis
Doctor visit schedule	1	2	High error rate (2/3 misclassified). 1 sample was misclassified as "Doctor practice hours" and another as "Online reservation patient list". This suggests semantic overlap and poorly labeled test sample
Medical procedure detail	2	1	1 sample was misclassified as "Consulted patient information". This suggests semantic overlap with patient consultation.
Medical service payment detail	2	1	1 sample misclassified as "Surgery schedule". This suggests semantic overlap when user query about surgery service payment.
Doctor feedback on chatbot usage	2	1	1 sample misclassified as "Medical service payment detail." This indicates that user feedback might contains terms which overlap with medical service intent.
Chatbot feature overview	0	3	Highest error rate (3 misclassified). This indicates that queries about the chatbot's features might contain terms which overlap with another intent.

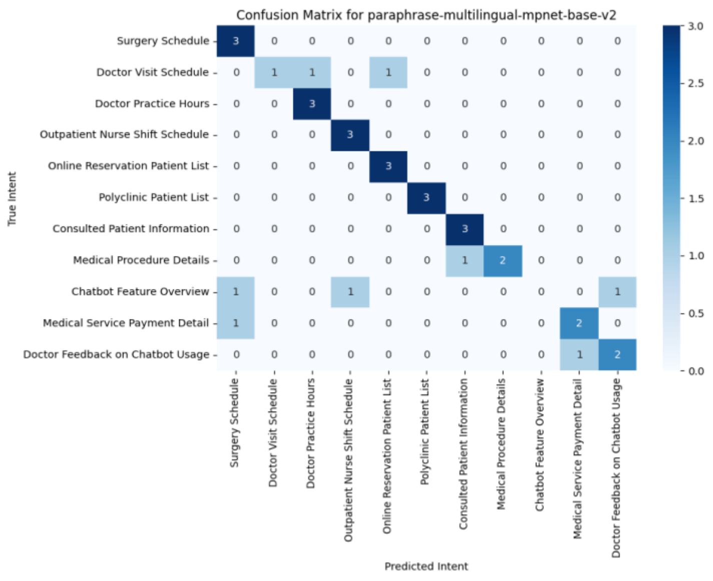


Figure 1. Confusion matrix

3.3. Semantic vector retrieval vs keyword matching comparison

We conducted an ablation study to compare the performance of the proposed semantic vector retrieval method against a traditional lexical search baseline for user intent recognition. The semantic vector retrieval utilized the best-performing embedding model, paraphrase-multilingual-mpnet-base-v2, combined with k=1 nearest neighbor classification using cosine similarity. The lexical baseline employed Jaccard similarity for keyword matching between the user query and the training intent samples. Both methods were evaluated using the accuracy metric on the intent test dataset. The results of the ablation study as shown in Table 6, demonstrate a clear superiority of the semantic vector retrieval approach in terms of recognition accuracy.

The semantic vector retrieval achieved an accuracy of 0.70. In contrast, the keyword search (lexical) baseline only reached an accuracy of 0.42. This result confirms that semantic vector retrieval provides a significant performance gain for user intent recognition [26], improving the accuracy by 28.0 percentage points. This improvement is attributed to the ability of the embedding model to capture the semantic meaning of the user's questions, which overcomes the limitations of simple keyword or token matching inherent in the Jaccard similarity method.

While the semantic vector retrieval method offered substantial accuracy improvements, it introduced a clear trade-off in processing time [27]. Keyword search was significantly faster, with an average latency of 13.622 ms per sample; it ran nearly 8x faster compared to semantic vector retrieval with an average latency of 109.043 ms per sample. This high efficiency is expected, as Jaccard similarity involves only simple string tokenization and set comparison, whereas semantic vector retrieval requires a computationally intensive model inference step [28].

Table 5. Misclassified test samples

True intent	Predicted intent	Original query (with English translation in parentheses)
Doctor visit schedule	Doctor practice hours	*Siapa saja pasien saya yang harus divisite hari ini* (Who are my patients scheduled for rounds/visits today)
Doctor visit schedule	Online reservation patient list	*Berapa jumlah pasien yang harus saya kunjungi sore ini* (What is the number of patients I need to visit this afternoon)
Medical procedure details	Consulted patient information	*Bagaimana prosedur permintaan konsul spesialis menurut panduan klinis* (What is the procedure for requesting a specialist consultation according to clinical guidelines)
Chatbot feature overview	Surgery schedule	*Apakah memiliki fitur untuk melihat jadwal operasi* (Does it have a feature to view the surgery schedule)
Chatbot feature overvie	Outpatient nurse shift schedule	*Apakah sistem bisa mengirim notifikasi ke perawat* (Can the system send notifications to the nurses)
Chatbot feature overview	Doctor feedback on chatbot usag	*Chatbot ini bisa menangani pertanyaan apa saja* (What kind of questions can this chatbot handle)
Medical service payment detail	Surgery schedule	*Berapa insentif yang saya terima dari tindakan operasi minggu lalu* (What is the incentive/remuneration I received for last week's surgical procedures)
Doctor feedback on chatbot usage	Medical service payment detail	*Saya tanya tentang jasa medis tapi jawabannya tidak jelas* (I asked about medical service fees but the answer was unclear)

Table 6. Comparison of semantic vs keyword search performance

Metric	Vector search (semantic, Cosine)	Keyword search (lexical, Jaccard)
Accuracy	0.70	0.42
Avg. latency (ms/sample)	109.043	13.622

3.4. Impact of similarity threshold on performance

The influence of the cosine similarity threshold (τ) on the semantic vector retrieval system's performance was assessed by plotting the precision and recall across a range of thresholds. The resulting curves, shown in Figure 2, demonstrate the classic precision-recall trade-off. As the threshold increases, the system becomes stricter, causing recall to decrease steadily, while precision initially rises before declining significantly at very high thresholds.

The most balanced performance, measured by the F1-score, is achieved where the two curves intersect or are closest. This optimal operating region was identified within the lower threshold range, specifically around τ=0.51. At this point, the system maximizes both the true intent capture rate (recall≈0.70) and the accuracy of those predictions (precision≈0.68). Crucially, thresholds above τ=0.70 resulted in a rapid collapse of recall, indicating that the system would frequently fail to classify legitimate user queries.

3.5. System design

HIMS is a complicated system that involves several users with diverse tasks. User-friendliness is a crucial component for the successful implementation of HIMS [29]. This paper concentrates on developing a chatbot application that fulfills a critical helpdesk function for doctors.

3.5.1. Chatbot general architecture

The general architecture of the chatbot is illustrated in Figure 3. Upon receiving a user inquiry, the system examines the request to ascertain the intended purpose (user intent). The subsequent activities and anticipated responses, which are contingent upon the identified user intent, can then be executed as potential solutions. Responses may be derived from several sources, such as an answer repository within the relational database management system (RDBMS), interaction with an external API, or the execution of SQL queries against the HIMS database. The response generation component transmits a reply to the user based on these obtained candidate responses. While the current system relies on predefined or retrieved data, a generative transformer model may be employed for future text generation to produce varied phrase forms [30], [31], enhancing conversational naturalness.

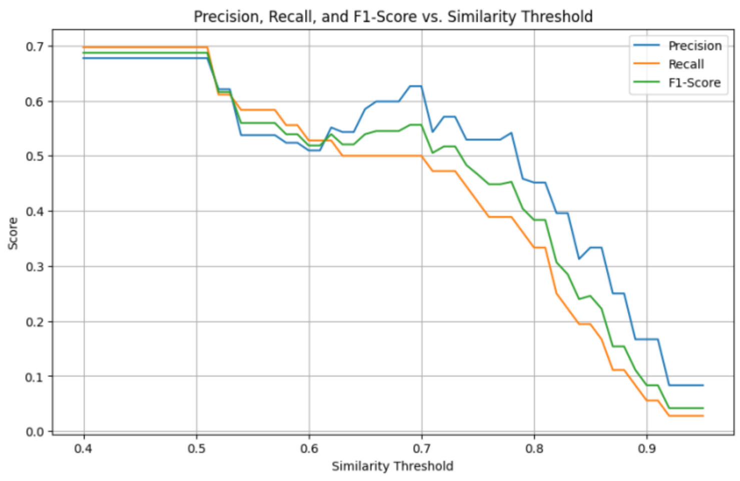


Figure 2. Threshold vs vector retrieval performance

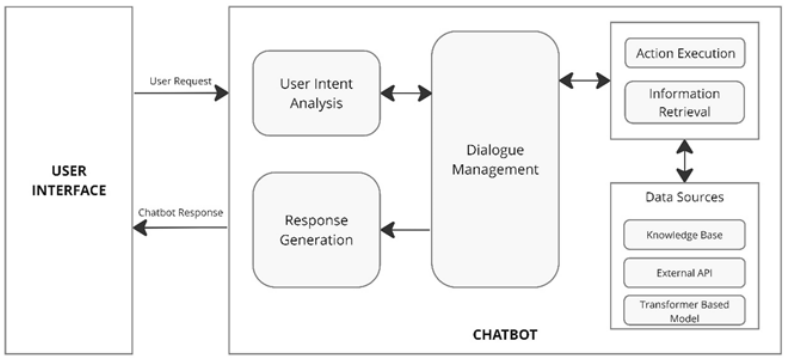


Figure 3. Chatbot general architecture

3.5.2. Semantic vector retrieval system architecture

Figure 4 illustrates the specific architecture used for the core intent recognition task. The system relies on a vector database management system (Milvus) and the central RDBMS. User intent definitions are transformed into feature vectors via a transformer-based deep learning model and maintained in the Milvus database. Upon receiving a user query, the transformer model converts the inquiry into a feature vector and performs a similarity vector retrieval within Milvus to identify the user intent most akin to the question. The identified user intent ID is then transmitted to the RDBMS to fetch the most suitable corresponding action or response.

3.6. System development and implementation

The chatbot application was developed using the Oracle Apex low-code platform for the user interface and quick application deployment [32]. This selection facilitates development speed and allows the application to be installed on smartphones using the progressive web apps (PWA) feature [33], [34]. Critical ML components are managed by a Python-based API [35] using the Django web framework, which performs inference and vector retrieval.

3.6.1. Semantic retrieval model and configuration

The deep learning model employed for vectorization is the paraphrase-multilingual-mpnet-base-v2 transformer model [14], which converts text into 768-dimensional feature vectors. For the vector storage and retrieval, the open-source Milvus vector database is utilized, employing a cosine similarity metric for the semantic vector retrieval [36]. The entire architecture, comprising the Oracle Apex server, the Django API, and the Milvus database, was deployed within a dedicated Google Cloud Platform (GCP) environment.

A sensitivity analysis was conducted to determine the optimal similarity threshold for deployment. Based on this analysis, the optimal threshold was determined to be 0.51. This value was selected because it maximizes the F1-score, representing the best operational balance between recall (the ability to retrieve all relevant intents) and precision (the accuracy of the retrieved intents). The chosen threshold of 0.51 ensures robust intent capture while maintaining high predictive reliability.

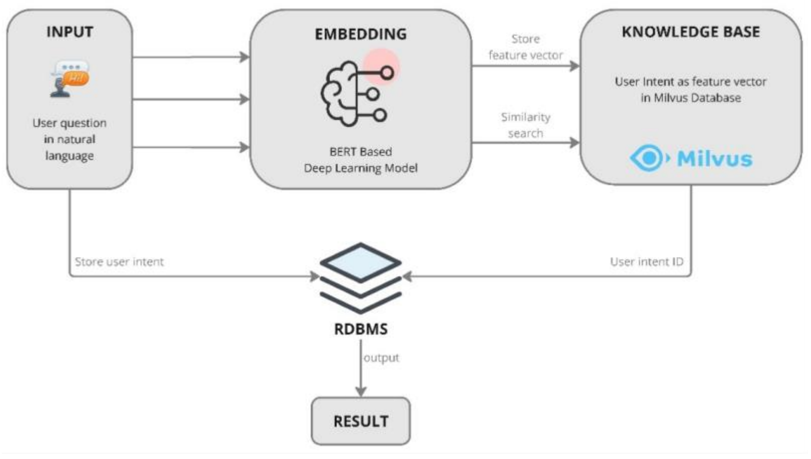
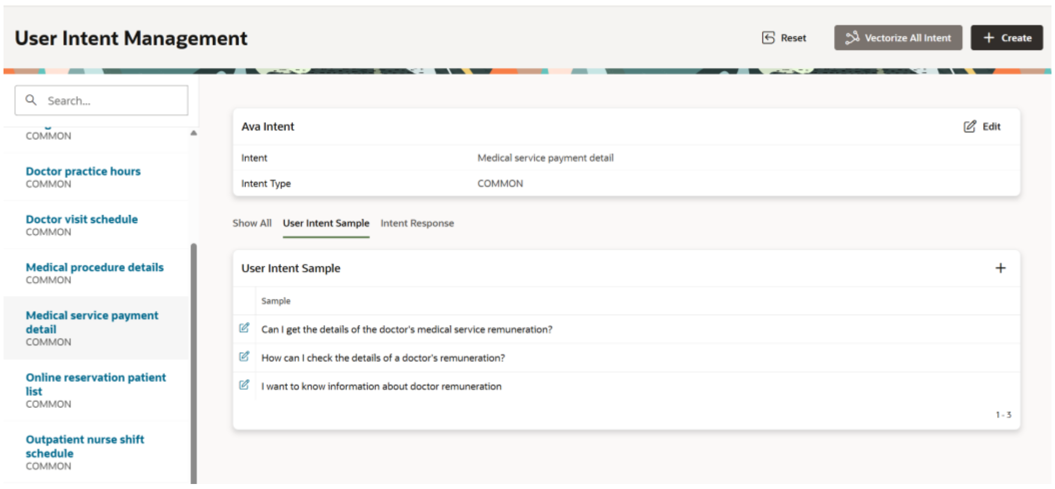


Figure 4. Semantic vector retrieval architecture

3.6.2. Chatbot application interfaces

The chatbot system includes both a management interface and a user interface. The management interface is used to oversee, update, and vectorize the knowledge base. It also serves as the interface for the AL loop, routing flagged, high-value user queries (identified via uncertainty metrics or user escalation) to experts for correction and labeling. The user interface is the platform for natural language interaction. It allows the user to inquire and receive responses based on the identified intent.

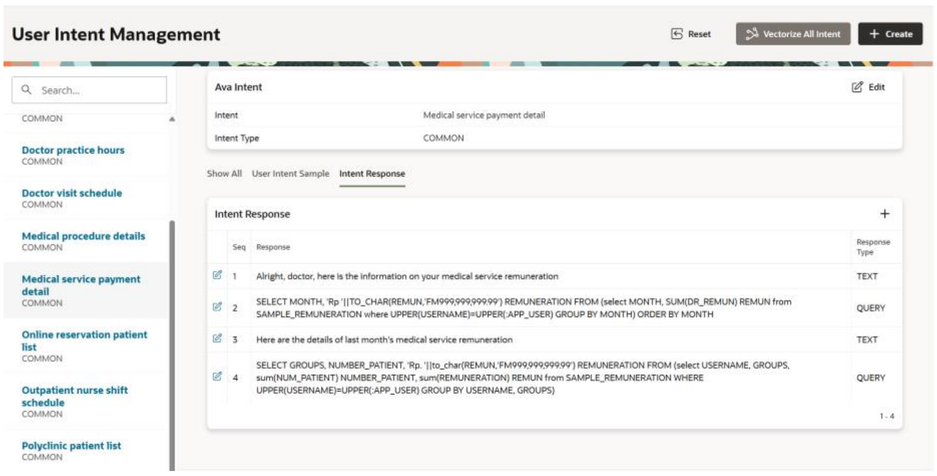
Management interface: the user intent management form (Figure 5) oversees intent definitions, request samples, and answers. The 'Vectorize all intent' button converts all intent sample data into feature vectors for storage in Milvus. The intent response interface (Figure 6) governs the response types (e.g., text, SQL query, API, URL, or button) and their sequence (Figure 7). User interface: the primary application interface (Figure 8) allows the doctor to input inquiries in natural language, demonstrating the system's ability to respond to requests articulated in a blend of Indonesian and English.



Intent	Medical service payment detail
Intent Type	COMMON

Sample
Can I get the details of the doctor's medical service remuneration?
How can I check the details of a doctor's remuneration?
I want to know information about doctor remuneration

Figure 5. User intent management: intent sample



Intent	Medical service payment detail
Intent Type	COMMON

Intent Response

Seq	Response	Response Type
1	Alright, doctor, here is the information on your medical service remuneration	TEXT
2	SELECT MONTH, 'Rp '||TO_CHAR(REMUN,'FM999,999,999.99') REMUNERATION FROM (select MONTH, SUM(DR_REMUN) REMUN from SAMPLE_REMUNERATION where UPPER(USERNAME)=UPPER(:APP_USER) GROUP BY MONTH) ORDER BY MONTH	QUERY
3	Here are the details of last month's medical service remuneration	TEXT
4	SELECT GROUPS, NUMBER_PATIENT, 'Rp. '||to_char(REMUN,'FM999,999,999.99') REMUNERATION FROM (select USERNAME, GROUPS, sum(NUM_PATIENT) NUMBER_PATIENT, sum(REMUNERATION) REMUN from SAMPLE_REMUNERATION WHERE UPPER(USERNAME)=UPPER(:APP_USER) GROUP BY USERNAME, GROUPS)	QUERY

1 - 4

Figure 6. Intent response

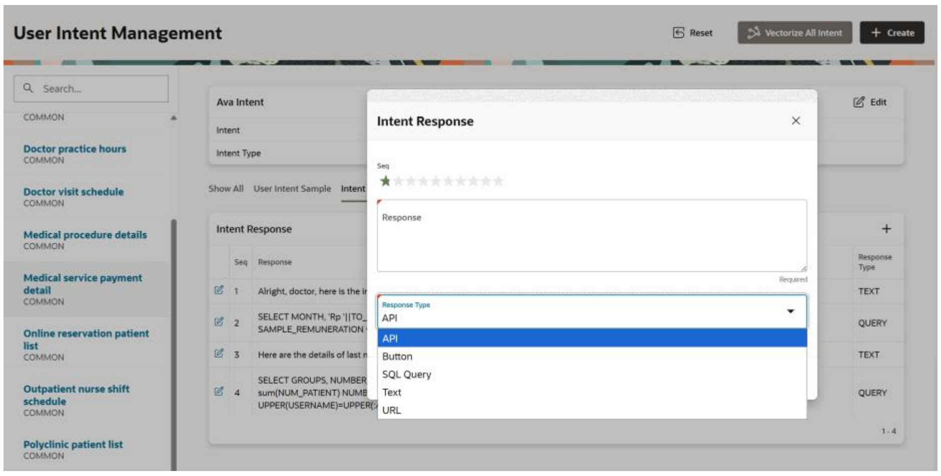
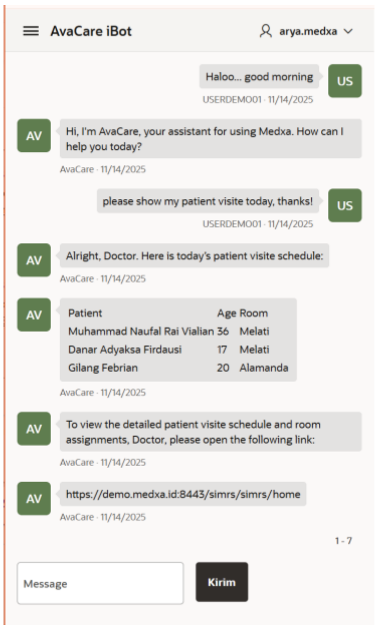


Figure 7. Intent response type



Haloo... good morning
USERDEMO01 · 11/14/2025

Hi, I'm AvaCare, your assistant for using Medxa. How can I help you today?
AvaCare · 11/14/2025

please show my patient visite today, thanks!
USERDEMO01 · 11/14/2025

Alright, Doctor. Here is today's patient visite schedule:
AvaCare · 11/14/2025

```
Patient                       Age Room
Muhammad Naufal Rai Vialian 36  Melati
Danar Adyaksa Firdausi        17  Melati
Gilang Febrian                20  Alamanda
```
AvaCare · 11/14/2025

To view the detailed patient visite schedule and room assignments, Doctor, please open the following link:
AvaCare · 11/14/2025

https://demo.medxa.id:8443/simrs/simrs/home
AvaCare · 11/14/2025

1 - 7

Figure 8. Chatbot application

3.7. Evaluation

Usability evaluation is conducted by the formulation of a questionnaire that encompasses efficiency, effectiveness, and satisfaction. Each facet has three questions, resulting in a total of nine questions pertaining to usability. A survey completed by the respondent after they use the chatbot application. A Likert scale ranging from 1 to 10 [24] was employed to assess respondents' level of agreement with statements about usability.

The nine questionnaire questions are part of the 38 usability testing attributes identified in a study by Nicole and Morgan [25]. Nine of the thirty-eight questions were chosen based on their relevance to the development of the helpdesk chatbot, since the study by Nicole and Morgan [25] aimed to assess conversational chatbots. The list of the nine questions can be seen in Table 7. The respondents were categorized into two groups: the HIMS development team, comprising 6 individuals, and doctors, totaling 15 individuals. The selection of doctors as respondents was due to the chatbot's awareness of the specified user intent, which aimed to address various queries from doctors.

Table 7. List of questionnaires

Aspect	Code	Question/statement
Efficiency	EI-01	Quick in giving response
Efficiency	EI-02	Able to handle unexpected requests
Efficiency	EI-03	Facilitates request escalation inside the ticketing procedure for human intervention.
Effectiveness	ES-01	Accurately interpret user requests
Effectiveness	ES-02	Easy to use
Effectiveness	ES-03	Able to provide convincing, satisfying, and natural interactions
Satisfaction	KP-01	Can ascertain the meaning or intent of a user's inquiry
Satisfaction	KP-02	Giving greetings, providing pleasant interactions
Satisfaction	KP-03	Providing a diverse reaction

The usability test respondents were carefully chosen from two distinct groups to provide a comprehensive evaluation perspective: 6 individuals from the HIMS application development team and 15 members of the doctor group. The HIMS development team, comprising software engineers and system architects, was selected due to their in-depth technical understanding of the system's underlying architecture and functionality. Their feedback is crucial for identifying technical usability issues and validating the system's adherence to design specifications. Their background might lead them to focus more on efficiency, performance, and technical robustness. Conversely, the doctor group, consisting of medical professionals who are primary end-users of the HIMS and the target users for this helpdesk chatbot, were chosen to evaluate the chatbot's practical utility, ease of use, and effectiveness in addressing their daily operational queries. To objectively determine if the observed differences in mean scores between the two groups were statistically reliable, an independent samples t-test (specifically Welch's t-test, due to unequal sample sizes) was conducted for each of the nine usability attributes. The t-test compared the mean ratings of the HIMS developer group (N=6) against the doctor group (N=15), with a significance level set at $p<0.05$.

The results of the questionnaire, administered to 21 respondents and illustrated in Figure 9, showed a mean score $\bar{x}=7.78$, which is interpreted as 'mostly agree' [37]. However, the independent samples t-test, with results detailed in Table 8, confirmed that none of these observed differences in mean scores were statistically significant ($p>0.05$) for any of the nine usability attributes. The overall mean scores, as shown in Table 9, indicated that the effectiveness aspect exhibited the highest mean value and efficiency the lowest in both respondent groups.

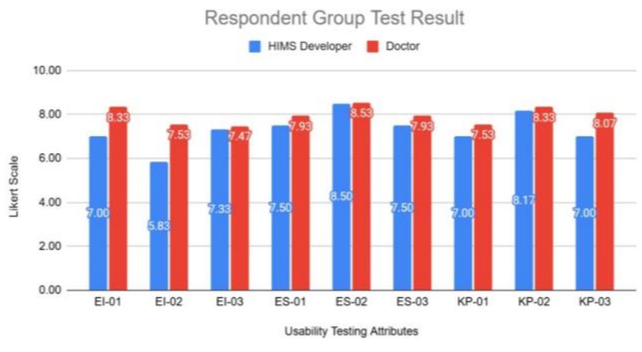


Figure 9. Test result

Table 8. Independent samples t-test

Attribute	Mean (developer)	Mean (doctor)	t-statistic	p-value	Significant (p<0.05)
EI-01	7	8.33	-1.4	0.209	No
EI-02	5.83	7.53	-1.35	0.229	No
EI-03	7.33	7.47	-0.17	0.869	No
ES-01	7.5	7.93	-0.71	0.5	No
ES-02	8.5	8.53	-0.06	0.954	No
ES-03	7.5	7.93	-1.3	0.213	No
KP-01	7	7.53	-0.98	0.348	No
KP-02	8.17	8.33	-0.26	0.798	No
KP-03	7	8.07	-1.84	0.091	No

Table 9. Questionnaire result

	HIMS developer	Doctor
Efficiency	6.72	7.78
Effectiveness	7.83	8.13
Satisfaction	7.39	7.98

Test points with a minimum test result of 8 or a verbal interpretation of ‘agree’ [37] are ES-02 and KP-02. The test points with the three lowest test results were EI-03, EI-02, and KP-01 with the lowest average value of 7.05 at test point EI-02. The doctors group gave higher average scores on all test points compared to the HIMS developer group.

To provide a deeper insight into the consistency of responses, the standard deviation for each aspect within both groups was also calculated. For the HIMS developer group (N=6), the standard deviations were: efficiency (≈1.526), effectiveness (≈0.349), and satisfaction (≈0.680). The very low standard deviation for effectiveness (≈0.349) indicates a strong consensus among developers regarding the chatbot's ability to interpret requests accurately, its ease of use, and its provision of convincing interactions. Conversely, the higher standard deviation for efficiency (≈1.526) suggests more varied opinions within this group concerning response speed and handling unexpected requests.

For the doctor group (N=15), the standard deviations were: efficiency (≈0.957), effectiveness (≈0.824), and satisfaction (≈1.306). While the mean scores for doctors were generally higher, the standard deviation for satisfaction (≈1.306) was the highest among all aspects and groups, indicating a notable range in doctors' overall satisfaction levels, particularly concerning the chatbot's ability to ascertain complex intent or provide diverse reactions. The efficiency aspect also showed more variability among doctors compared to effectiveness. This detailed view of response spread complements the mean scores by highlighting areas of higher or lower agreement within the respondent groups.

3.8. Computational performance and load testing

The system performance evaluation results demonstrate that the Milvus vector database maintained low query latency even under high concurrency (Table 10). At single-user operation, the average latency was approximately 13.6 ms per query, and at 100 concurrent users, latency increased only modestly to 21.4 ms, indicating stable scalability. Throughput increased linearly with concurrent load, reaching over 6,000 queries per second at 100 simultaneous users. This indicates that Milvus can efficiently handle real-time semantic retrieval workloads in healthcare chatbot environments, ensuring both responsiveness and reliability during peak interaction periods.

Table 10. Performance and load testing result

Concurrent users	Avg. latency (ms)	P95 latency (ms)	Throughput (req/sec)
1	13.6	15.2	73.5
5	14.8	17.3	365.2
10	15.9	19	715.8
25	17.2	21.6	1750.3
50	18.8	24.9	3290.1
100	21.4	28.6	6120.4

3.9. Limitations and challenges

While the developed chatbot system demonstrates promising capabilities, several limitations and challenges were identified during its development and deployment: while semantic vector retrieval significantly improves natural language understanding, the chatbot's ability to handle highly ambiguous queries remains a challenge. The effectiveness of the semantic vector retrieval is directly tied to the comprehensiveness and quality of the 'Intent Samples' in the Milvus database. If a user's query deviates significantly from the trained samples, the system might provide a less accurate or irrelevant response.

The accuracy and relevance of the chatbot's responses are directly dependent on the 'User intent' and 'Intent response' data. Regularly updating and expanding this knowledge base, especially as HIMS functionalities evolve or new medical queries arise, requires dedicated effort and a structured content management process. The current semantic vector retrieval architecture is inherently a single-turn, task-based retrieval system. It is unable to maintain conversational context across multiple turns or handle complex follow-up questions (e.g., "What is my schedule?" followed by "Now, what about the nurse's schedule?"), limiting its capability for natural, sustained dialogue.

As the number of users and the complexity of user intents grow, maintaining optimal response times and ensuring the efficient functioning of the vector database and transformer model can become a scalability challenge. Future scaling might require more sophisticated load balancing or distributed database architectures. The suggestion to increase server specifications by adding a GPU highlights an anticipated need for higher computational resources as demand increases.

4. CONCLUSION

This research successfully designed and developed a healthcare chatbot application to assist the HIMS helpdesk system by leveraging vector database technology, demonstrating its capability to recognize complex user intents in natural language. The crucial finding from the ablation study confirmed the efficacy of the semantic approach: vector search achieved an intent recognition accuracy of 0.70, representing a significant 28.0 percentage point gain over the traditional keyword-based search baseline. This improvement is directly attributed to the MPNet-based model's ability to capture the semantic meaning of user queries, overcoming the limitations of simple token matching. While this high accuracy introduced a performance trade-off, running nearly 8x slower than the lexical baseline, the gain in understanding justifies the adoption of this robust architecture. The overall usability of the chatbot was validated through a questionnaire administered to both technical developers and primary end-users (doctors), yielding a grand mean score $\bar{x}$=7.78 on a 10-point scale, interpreted as 'mostly agree'. Crucially, statistical analysis confirmed a consistent assessment between the two respondent groups, as no statistically significant difference was found in their mean usability ratings. However, three key areas for improvement were identified by the lowest scoring test points: the ability to 'handle unexpected requests', the capacity to 'ascertain the meaning or intent of a user's inquiry', and the functionality to 'facilitate request escalation'. Building upon this foundational work, several avenues for future system improvements are envisioned. To address the challenge of fixed vector boundaries and ambiguous queries, we propose implementing an AL loop to continuously refine the user intent knowledge base by incorporating flagged, high-value user queries into the Milvus vector database. Furthermore, future research should explore utilizing more advanced transformer models capable of maintaining conversational context across multiple turns and expanding the chatbot's capability to integrate and synthesize information from a wider range of real-time HIMS data feeds. Addressing these areas will ensure the system provides increasingly accurate, comprehensive, and natural 24-hour assistance.

FUNDING INFORMATION

The authors declare that no funding was received for conducting this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Erda Guslinar Perdana	✓	✓		✓	✓			✓		✓		✓		
Arya Adhi Nugraha	✓	✓	✓			✓	✓		✓		✓			

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [EGP], upon reasonable request.

REFERENCES

[1] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," *Artificial Intelligence Applications and Innovations*, 2020, pp. 373–383, doi: 10.1007/978-3-030-49186-4_31.

[2] S. R. Dammavalam, N. Chandana, T. R. Rao, A. Lahari, and B. Aparna, "AI based chatbot for hospital management system," in *2022 3rd International Conference on Computing, Analytics and Networks*, Nov. 2022, pp. 1–5, doi: 10.1109/ICAN56228.2022.10007105.

[3] D. A. R. Lede *et al.*, "Tana, a healthcare chatbot to help patients during the COVID-19 pandemic at a University Hospital in Argentina," *Studies in Health Technology and Informatics*, vol. 6, no. 290, pp. 301-303, 2022, doi: 10.3233/SHTI220083.

[4] W. Maeng and J. Lee, "Designing a chatbot for survivors of sexual violence: exploratory study for hybrid approach combining rule-based chatbot and ML-based chatbot," in *Asian CHI Symposium 2021*, May 2021, pp. 160–166, doi: 10.1145/3429360.3468203.

[5] S. A. Thorat and V. Jadhav, "A review on implementation issues of rule-based chatbot systems," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3567047.

[6] Y. Bang *et al.*, "A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 2023, pp. 675–718, doi: 10.18653/v1/2023.ijcnlp-main.45.

[7] H. Alkaissi and S. I. McFarlane, "Artificial hallucinations in ChatGPT: implications in scientific writing," *Cureus*, vol. 15, no. 2, Feb. 2023, doi: 10.7759/cureus.35179.

[8] W. X. Zhao, J. Liu, R. Ren, and J. R. Wen, "Dense text retrieval based on pretrained language models: a survey," *ACM Transactions on Information Systems*, vol. 42, no. 4, 2024, doi: 10.1145/3637870.

[9] Y. Park and Y. Shin, "Adaptive bi-encoder model selection and ensemble for text classification," *Mathematics*, vol. 12, no. 19, Oct. 2024, doi: 10.3390/math12193090.

[10] J. Choi, E. Jung, J. Suh, and W. Rhee, "Improving bi-encoder document ranking models with two rankers and multi-teacher distillation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2021, pp. 2192–2196, doi: 10.1145/3404835.3463076.

[11] S. S. Chen, "A 'code-switching' model for healthcare communication," *Healthcare Management Forum*, vol. 38, no. 4, pp. 391–394, Jul. 2025, doi: 10.1177/08404704251327095.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: sentence embeddings using siamese BERT-networks," *EMNLP-IJCNLP 2019-2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing,* pp. 3982–3992, 2019, doi: 10.18653/v1/D19-1410.

[13] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers," *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020, pp. 1-13.

[14] K. Song, X. Tan, T. Qin, J. Lu, and T. Y. Liu, "MPNet: masked and permuted pre-training for language understanding," *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020, pp. 1-11.

[15] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: massive text embedding benchmark," *EACL 2023-17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2006–2029, 2023, doi: 10.18653/v1/2023.eacl-main.148.

[16] A. Andoni, P. Indyk, and I. Razenshteyn, "Approximate nearest neighbor search in high dimensions," *Proceedings of the International Congress of Mathematicians, ICM 2018*, vol. 4, pp. 3305–3336, 2018, doi: 10.1142/9789813272880_0182.

[17] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2020, doi: 10.1109/TPAMI.2018.2889473.

[18] J. Wang *et al.*, "Milvus: a purpose-built vector data management system," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Jun. 2021, pp. 2614–2627, doi: 10.1145/3448016.3457550.

[19] M. Buckmann, E. Hill, and A. Analytics, "Logistic regression makes small LLMs strong and explainable 'tens-of-shot' classifiers," *arXiv:2408.03414*, 2024.

[20] A. Mishra and Y. I. Alzoubi, "Structured software development versus agile software development: a comparative analysis," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 4, pp. 1504–1522, Aug. 2023, doi: 10.1007/s13198-023-01958-5.

[21] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: review and analysis," *VTT Publications*, no. 478, pp. 3–107, 2002.

[22] S. Kass, S. Strahringer, and M. Westner, "Practitioners' perceptions on the adoption of low code development platforms," *IEEE Access*, vol. 11, pp. 29009–29034, 2023, doi: 10.1109/ACCESS.2023.3258539.

[23] R. Ren, M. Zapata, J. W. Castro, O. Dieste, and S. T. Acuna, "Experimentation for chatbot usability evaluation: a secondary study," *IEEE Access*, vol. 10, pp. 12430–12464, 2022, doi: 10.1109/ACCESS.2022.3145323.

[24] A. Joshi, S. Kale, S. Chandel, and D. Pal, "Likert scale: explored and explained," *British Journal of Applied Science & Technology*, vol. 7, no. 4, pp. 396–403, Jan. 2015, doi: 10.9734/bjast/2015/14975.

[25] R. Nicole and B. Morgan, "Evaluating quality of chatbots and intelligent conversational agents," *CEUR Workshop Proceedings*, vol. 1982, pp. 40–49, 2017.

[26] A. Ahluwalia, B. Sutradhar, K. Ghosh, I. Yadav, A. Sheetal, and P. Patil, "Hybrid semantic search: unveiling user intent beyond keywords," *arXiv:2408.09236*, 2024.

[27] L. Harris, "Comparing lexical and semantic vector search methods when classifying medical documents," *arXiv:2505.11582*, Jun. 2025.

[28] A. M. S, S. Mangrulkar, and V. Sembium, “HISS: a novel hybrid inference architecture in embedding based product sourcing using knowledge distillation,” *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom’22)*, vol. 1, 2022.

[29] A. Setiorini, S. R. Natasia, Y. T. Wiranti, and D. A. Ramadhan, “Evaluation of the application of hospital management information system (SIMRS) in RSUD Dr. Kanujoso Djatiwibowo using the HOT-Fit method,” *Journal of Physics: Conference Series*, vol. 1726, no. 1, Jan. 2021, doi: 10.1088/1742-6596/1726/1/012011.

[30] T. Xue, Y. Zhao, C. Yang, G. Liu, and X. Li, “SECT: a successively conditional transformer for controllable paraphrase generation,” in *Proceedings of the International Joint Conference on Neural Networks*, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892042.

[31] A. Singh and G. S. Josan, “Paraphrase generation: a review from RNN to transformer based approaches,” *International Journal of Next-Generation Computing*, vol. 13, no. 1, Apr. 2022, doi: 10.47164/ijngc.v13i1.377.

[32] Pique Solutions, *Oracle application express (APEX): a time and motion analysis developing applications using APEX versus traditional development approaches*, Sans Fransisco, California, 2020.

[33] M. S. Rao, “Progressive web apps in cross-platform development: a comparative analysis and evaluation,” *Interantional Journal of Scientific Research in Engineering and Management*, vol. 8, no. 6, pp. 1–5, Jun. 2024, doi: 10.55041/ijsrem35595.

[34] O. Jerome C, A. Onyekachi A., A. Bethran C, and A. VC, “Effective cross-platform mobile app development using progressive web apps, deep learning and natural language processing,” *International Journal of Engineering Applied Sciences and Technology*, vol. 7, no. 9, pp. 14–20, Jan. 2023, doi: 10.33564/ijeast.2023.v07i09.003.

[35] P. Thakur and P. Jadon, “Django: developing web using Python,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2023*, May 2023, pp. 303–306, doi: 10.1109/ICACITE57410.2023.10183246.

[36] M. Al Sulaiman, A. M. Moussa, S. Abdou, H. Elgibreen, M. Faisal, and M. Rashwan, “Semantic textual similarity for modern standard and dialectal Arabic using transfer learning,” *PLoS ONE*, vol. 17, no. 8 August, Aug. 2022, doi: 10.1371/journal.pone.0272991.

[37] H. Taherdoost, “What is the best response scale for survey and questionnaire design; review of different lengths of rating scale/attitude scale/Likert scale,” *International Journal of Academic Research in Management*, vol. 8, no. 1, pp. 1–10, 2019.

BIOGRAPHIES OF AUTHORS

Erda Guslinar Perdana obtained his doctoral degree and master's degree from the Bandung Institute of Technology (ITB) in Indonesia in 2023 and 2012, respectively. He earned his bachelor's degree from Telkom University in 2007. Currently, he serves as a lecturer in Software Engineering Applications Study Program, School of Applied Science, Telkom University, Indonesia. His research interests are focused on enterprise architecture and applied artificial intelligence. He’s a member of Center of Excellence for Inspiring Digital Transformation for Social Innovation (InsPiRo), Research Institute of Sustainable Society, Telkom University. He can be contacted at email: erda@telkomuniversity.ac.id or erda.guslinar@gmail.com.

Arya Adhi Nugraha is the Chief Technology Officer (CTO) at Medxa, where he leads the development and implementation of cutting-edge healthcare technology solutions. With extensive experience in hospital information systems (HIMS) and AI-driven healthcare applications, he specializes in integrating machine learning models into clinical decision support, hospital administrative tasks, and electronic health record interoperability. His research interests include AI-driven diagnostics, computer vision for healthcare, natural language processing (NLP) for healthcare, and regulatory compliance in digital health. He holds a Bachelor’s degree in Information System from Indonesia Open University. He can be contacted at email: arya.adhi@medxa.co.id.