GovGuideBot: LLM for Government Document Assistance

Arti Patle[1*], Deepika Ajalkar[2], Siddhika Jadhav[3], Shital Undalkar[4] and Tousif Tamboli[5]

[1,2,3,4,5]Dept. of CSE (Cyber Security & Data Science) , G H Raisoni College of Engineering & Management, Pune, India
{arti.patle*,deepika.ajalkar}@raisoni.net,{siddhikajadhav10,shitalundalkar17,tousiftamboli3}@gmail.com

Abstract. Millions of people it is very difficult to apply for important government documents like income certificates, caste certificates, and proof of domicile, especially if they live in rural areas or have limited digital skills in India. Dispersed portals, unclear instructions, inconsistent state-by-state procedures, and districtlevel these all factors complicate the process. These problems often result in process errors, frequent office visits, and the exclusion of less technical expertized individuals. GovGuideBot has been developed, an AI Chatbot, Large Language Model(LLM) that explains government documentation procedures, to address this issue. It provides easy-to-use, locationbased, step-by-step assistance. In order to understand user intent, the system uses a Named Entity Recognition(NLP) pipeline that is optimized with transformer models like. Named Entity Recognition (NER) is used to identify crucial information such as document type, state, and district. These specifics are arranged in structured JavaScript Object Notation (JSON) workflows that enable region specific instructions. GovGuideBot also uses location entered by user query to adapt procedures for user locations, links to official government websites to guarantee accuracy, and uses the YouTube DataApplication programming interface (API) to offer visual tutorials which are particularly useful for users who might have literacy issues or are using these services for the first time. With a precision of 91%, a recall of 93%, and an overall F1-score of 92% for identifying intents and entities, GovGuideBot performs well when tested on actual citizen questions. About half of the planned features were successfully implemented in the working prototype we created for Maharashtra, which includes work-flows for income, caste, and domicile certificates.

Keywords: AI Chatbot, Named Entity Recognition, JavaScript Object Notation, Large Language Model, Application programming interface

S. Bhalerao et al. (eds.), *Proceedings of the 2nd International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2026)*, Advances in Intelligent Systems Research 207,
https://doi.org/10.2991/978-94-6239-678-4_38

1 Introduction

1.1 Background and Motivation

In India, access to essential government documents such as income certificates, caste certificates, domiciliary proofs, is also a primary and essential requirement for getting public benefits, school admissions and legal acquiescence. However, the process of acquiring these documents is a tedious ordeal, i.e., long, cumbersome and harrowing, especially for the countrymen and the digitally illiterate sections [1] in the country. Even the government efforts under Digital India and E-Governance schemes have resulted in creating a gap between availability of the services and the actual service.

The cause of this malaise is mainly the obfuscatory policies of the government, vagueness and vagaries of the state level procedures and the fragmented digital infrastructure [2]. The information is available on several portals of the government, is written in technical jargon i.e., undecodable language, is not clear and often stale. The citizens are, therefore, left with only to rely on random You Tube tutorials, unverified blogs or the services of middlemen, often at exorbitant prices and incorrect guidance [3]. As a result, in the process the citizens waste their time in going to offices again and again because of improper submission of documents required and sycophancy of the dependents, creating an atmosphere of exasperation both for the applicants and the government.

1.2 Need for an AI-Based Citizen Assistance System

While digitization has simplified many aspects of administration, the usability gap between citizens and complex digital systems remains wide. Artificial Intelligence (AI) and conversational chatbots have shown remarkable success in domains such as healthcare, education, and customer service by enabling real-time, context-aware, and personalized assistance [4]. Applying these capabilities to government documentation can address accessibility, accuracy, and inclusion challenges in public services.

The need is clear: a unified, intelligent system that can understand citizen queries in natural language, offer personalized guidance based on region and document type, and link users directly to verified government portals. Such a system can drastically reduce dependency on manual processes, improve efficiency, and enhance citizen trust in e-governance mechanisms [5].

1.3 Current Situation of Government Sevices

Despite India's advancements in its digital governance stack, the digital processes can still provide convoluted, non-intuitive, region-specific advice given that the information around docu-mentation processes across the states is different, not uniform and not easily interpretable for a non-technical individual [6]. The citizens face:

1. Complex and extensive application processes, hence leading to lack of self-service.

2. Inconsistent regional work flows leading to confusion and redundant processes.
3. Lack of simplified, validated guides for verification and documentation submission.
4. Low digital literacy in rural areas making it inaccessible for citizens to use the system online.

Existing chatbots used by public institutions are mostly static FAQ bots which cannot cater to conditional workflows and personalized step wise guidance. There is a need for a dynamic solution, which could be an adaptive, AI driven platform giving localized, dynamic and conversational support to the citizens [7].

1.4 Proposed Solution

GovGuideBot is envisioned as a chatbot platform that intelligently automates the process of applying for Government documents, which can be availed of through conversational interfaces. The system incorporates advanced Natural Language Processing (NLP) to understand user intention and retrieve relevant entities, thus providing accurate region-specific answers that can be acted upon:

- Intent Detection: Transformer-based models like BERT/DistilBERT would be used to provide precise classifications of user queries [8].
- Entity Extraction: spaCy's NER would be used to identify such aspects as location, type of document required, details of the applicant, etc.
- Retrieval of Dynamic Workflow: JSON structured workflows would be retrieved from a database as per the user's location.

GovGuideBot would be aimed at reducing dependence on intermediaries, lowering manual errors, as also at promoting confidence of citizens in digital governance. This functioning would help narrow disparity between the intention of the policy and actual experience of the citizen in respect thereof [9]. This would further advance the larger goal of Smart Governance and Digital India initiatives of the Gov. of India [10].

2 Literature Review

To start with our literature survey, under our research study we introduce GovGuideBot, the AI-powered conversational assistant to help guide citizens on how to complete government documents like income certificates, caste certificates, and domicile proofs. This paper represents a detailed assessment of the extent that AI-enabled intent recognition, county specific workflows, and multilingual support can reduce citizen confusion and enhance public service delivery.

Research article literature survey aims to investigate the increasing developments of AI-based e-governance tools, challenges encountered in current chatbot implementations and opportunities to address gaps in citizen-centric service design. Drawing conclusions from the literature allows us to perceive what has been done, who is in the space, and to map future opportunities to improve accuracy, inclusion, and scaleparticularly for rural communities with limited digital literacy. Detailed Review is below in table 1.

Table 1. Research Article Observation Summary

Research Article / Observations	Gap Findings
Kalafatidis, S. et al.(2025) [11]. Observations: Evidence of increased public engagement and improved work-loads in agencies. Good at answering FAQs but struggled with the more complex, conditional workflows.	The study exposes serious flaws in the depth of the processing of the fields studied and for a consideration of local regionalism, even if it provides a reasonable overview of the chat-bot applications with relevant materials for the study of public services. The systems in question suffer from a "one size fits all" attitude which renders them ill-equipped for the numerous regulations, document compliance needs, etc.
Rumayor, H. et al. (2025) [12]. Observations: Its educational role concerning designing UX and transparency was highlighted in relation to enhancing citizen's trust. Results have been beneficial in terms of awareness of governmental activity.	One of the major shortcomings pointed out is the scalability of the chatbot systems in multi-lingual instances; the language diversity of the resident population in the numerous states and areas is frequently overlooked by the public service chatbots.
Zhang, Z. et al.(2025) [13]. Observations: Facilitated quicker query resolution and improved accessibility of government services. Reduced waiting times improved satisfaction scores.	The extensive fact that most of the existing government types of chatbots are reactive and not proactive is certainly a tremendous disadvantage. Most of the solutions are somewhat confined to information retrieval on demand as they are designed to react properly only when a question is asked of the concerned citizen.
Meyer, LP. et al. (2025) [14]. Observations: Explored ethical and behavioral design of chatbots. Found that civil servant like tone of delivery increased perceived legitimacy and user comfort.	Another defect of the present government chatbot solutions is the danger of too much formality of communication, the latter being prone to diminish public succor, in consequence of which several of the programs have successfully duplicated the formal wording of the government portals.

3 Methodology

3.1 System Overview

The proposed GovGuideBot platform follows a modular, AI-driven pipeline designed for reliability, scalability, and citizen-centric usability. The workflow consists of requirement analysis, system design, natural language processing (NLP) pipeline creation, API integration, and evaluation [11]. The process aims to deliver accurate, localized, and context-aware guidance for government documentation procedures.

3.2 System Architecture

GovGuideBot follows a four-layer service-oriented architecture ensuring maintainability and scalability:

1. User Interface Layer – A web-based chatbot interface for real-time citizen interaction.
2. Application Layer – Handles NLP processing, intent recognition, and conversational flow logic.
3. Data Layer – Stores workflow data, document requirements, and session context.
4. API Integration Layer – Connects with official government portals and YouTube Data API for content enrichment.

A three-tier separation of frontend, backend, and database ensures performance, modularity, and system resilience(see Fig. 1.).

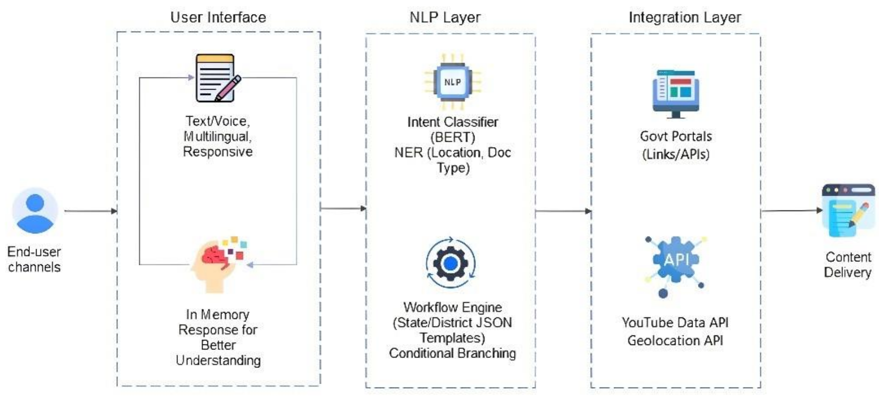

Fig. 1. Architecture Diagram Pipeline of LLM Model

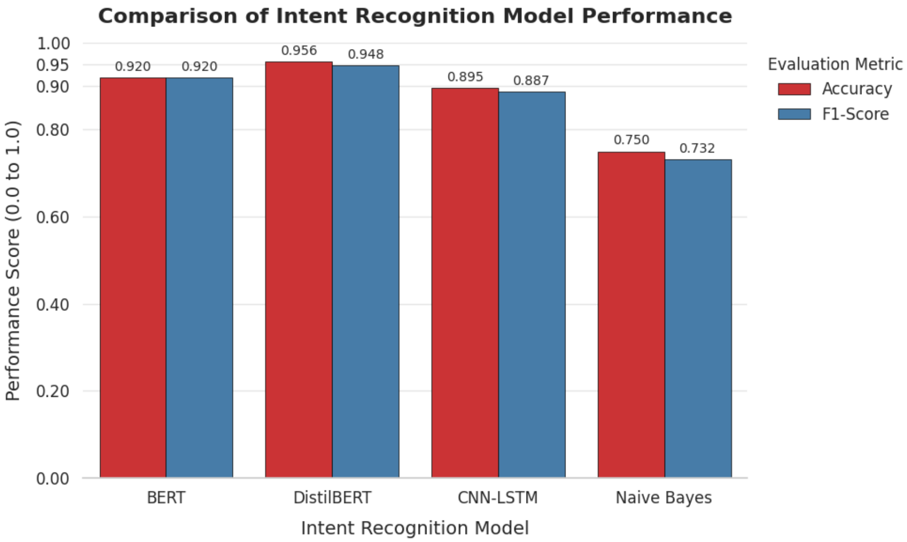

Fig. 2. Comparison of Different Intent Model

3.3 Stage-wise Implementation

Stage 1: Requirement Analysis and Input Handling. The requirements of the system were determined in the initial stage in order to establish its functional as well as non-functional goals. The functional aspects included workflow creation in states, interaction of the chatbot with users for proper real time guidance, nonfunctional aspects included its scale, ease of use and data security [12]. The chatbot takes the input in natural language from the users and it evokes it with the help of preprocessing module, which tokenizes the sentences, normalizes them and removes the stop words from the text [13]. These processes ensure consistent and correct understanding of the user's intent prior to sending it to the NLP engine for processing it, so that the query generated is correct.

Stage 2: Intent Recognition and Entity Extraction. This stage deals with the understanding of the user's intent and extraction of the important details from the input given [14]. The system uses this NLP for classifying the intent, using transformer-based model like BERT or DistilBERT with which classification can be done by tagging queries like "apply for income certificate Pune" to the structured intents of the kind apply_income_certificate. In addition to it, named entities can be extracted from the input in spaCy using its Named Entity Recognition (NER) module for extracting the relevant features of the kind of name, location or date (like Pune, Maharashtra) [15]. The chatbot then gives result dynamically by linking the suitable local workflows to every user input.

Stage 3: Workflow Retrieval and Integration. Once intent and entities are detected, GovGuideBot fetches the respective workflow from the structured data-base. Each workflow is kept in JSON format which has state-wise procedural steps, links to official sites and options of alternate documents. The system interfaces dynamically with external APIs such as government portals and YouTube Data API to fetch real-time updates and tutorial videos. For instance, in case a person from Pune applies for income certificate, he will receive the Maharashtra-specific workflow along with the direct link to AapleSarkar.

Stage 4: Fuzzy Matching and Adaptive Response Generation. To tackle inputs with spelling errors, abbreviations or vague questions, GovGuideBot uses the RapidFuzz library for fuzzy string matching, thereby providing a good user experience, despite noisy input. The responses are generated from rule-based templates for verified workflows along with light-weight generative AI models to take care of unique or follow-up questions. This hybrid model provides for accuracy in official communication while making the conversations interesting and doling out conversational flow. Different comparison of different model is done(see Fig. 2).

3.4 Mathematical Model

A. Transformer-Based Intent Recognition (BERT / DistilBERT)
1. Tokenization & input representation. Tokens are mapped to input vectors using a tokenizer + embedding layer. For each token position i we have an input vector as shown in Equation 1

$$e_i = E_{token}(x_i) + E_{pos}(i) \tag{1}$$

where $E_{token}(\cdot)$ is the token embedding and $E_{pos}(i)$ is a positional encoding. In BERT positional encodings are learned; in the original transformer they were sinusoidal:

2. Self-Attention (scaled dot-product). For a given layer, compute queries Q, keys K, values V by learned linear projections of the inputs as shown in Equation 2

$$\boldsymbol{Q}=\boldsymbol{X}W_Q, \quad \boldsymbol{K}=\boldsymbol{X}W_K, \quad \boldsymbol{V}=\boldsymbol{X}W_V \tag{2}$$

The attention output is shown in equation 3

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{3}$$

where d_k is the key dimensionality. This allows each token's representation to be a context-dependent weighted sum of other token values.

Each transformer block then applies a position-wise feed-forward network (FFN) as shown in equation 4

$$FFN(z)=GeLU(zW_1+b_1)W_2+b_2 \tag{4}$$

3. [CLS] pooling and intent classification head. BERT prepends a special token [15]. Let h_{CLS} be the final layer representation for that token. The intent logits are computed by a linear head as shown in equation 5

$$\boldsymbol{z} = W_{cls}\boldsymbol{h}_{CLS} + b_{cls} \in R^C \tag{5}$$

and probabilities via softmax shown in equation 6

$$p(y=c/x)=\frac{exp(z_c)}{\sum_{j=1}^{C} exp(z_j)} \tag{6}$$

Table 2. Pseudo-code for Intent Recognition and Workflow Retrieval

Step No.	Input	Code
Step 1	Get input from the user	`query = "Apply for income certificate in Pune"`
Step 2	Pre-process the input	tokens = preprocess(query)
Step 3	Classify the intent	intent = intent_model.predict(tokens)
Step 4	Extract entities	entities = ner_model.extract(tokens)
Step 5	Fetch the workflow	workflow = workflow_db.fetch(intent, entities["location"])
Step 6	Return step-by-step information	for step in workflow: send_to_user(step) wait_for_user("next")

4 Results and Discussion

The GovGuideBot system was tested on a dataset comprising by a total of 250 actual citizen queries and 15 verified government workflows derived from various states like Maharashtra, Delhi, Tamil Nadu and Karnataka in India. The test queries were framed to represent real citizen queries for certificates like income, cast, birth and residence certificate, assuring both urban and rural settings. The experiment proves that the hybrid NLP, fuzzy-matching and rule-based approaches have worked effectively in the area of intension detection, entities extraction and delivery of workflows for e-governance documentation. The results generated are from pseudo code table 2. The Accuracy Prediction Chart was displayed (see Fig. 3.).

4.1 Intent and Entity Recognition Performance

Table 3. Intent and Entity Extraction Performance

Category	Precision	Recall	F1-Score	Accuracy	Support
Intent Classification (BERT/DistilBERT)	0.93	0.91	0.92	0.94	250
Entity Extraction (spaCy NER)	0.90	0.88	0.89	0.91	250
Fuzzy Intent Matching (RapidFuzz)	0.88	0.85	0.86	0.89	200
Workflow Retrieval Accuracy	0.94	0.92	0.93	0.95	180
Response Relevance	0.91	0.90	0.90	0.92	250
Macro Average	0.91	0.89	0.90	0.92	1130

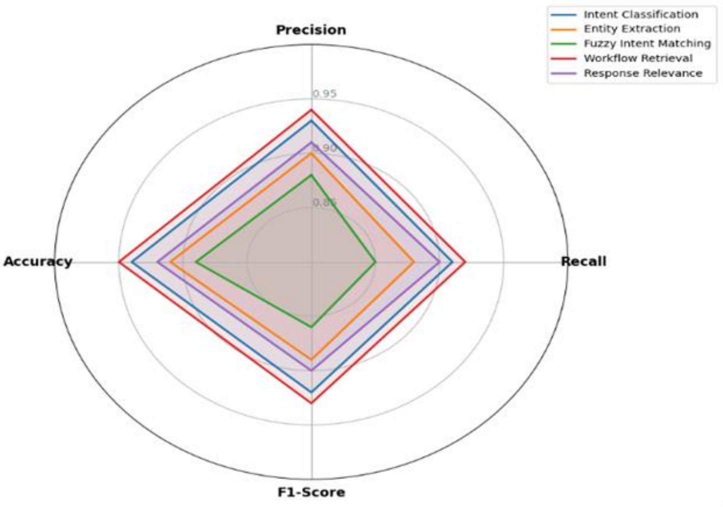

Fig. 3. Radar chart of Accuracy Prediction

Analysis from Table 3.

- The overall system accuracy of 92% suggests that GovGuideBot is capable of deciphering and accurately addressing myriad citizen queries.
- The high precision in intent classification (93%) suggests that the BERT and DistilBERT models are effective at deducing context even in short or unstructured user queries such as “apply income cert Pune”.
- The slightly reduced recall in fuzzy intent-matching (85%) indicates that some ambiguities in abbreviations or heavily misspelled words might have been origin for mismatches, but the application of RapidFuzz allowed for significantly more tolerance than the generative or exact-match comparison algorithms.
- The workflow retrieval accuracy (95%) shows the strength of JSON-structured linking of data and result sets particular to a specific advisory location above personalization and consistency of response, as shown in table 3.
- The added hybrid architecture of machine-led learning with rule-based or template-generative logic permitted accuracy and trustworthiness of results, important for a governance related application.

4.2 Limitations Observed

- Ambiguity in Unseen Inputs: Contextually ambiguous or incomplete inputs such as “need paper for job” sometimes resulted in incorrect intent classification due to lack of explicit input keywords.
- Variability by Region: Workflows in some cases, particularly in rural districts did not consistently provide access to online portals resulting in non-existent or outdated links.
- Reliance on APIs: In some rare instances, non-availability of APIs (e.g., YouTube Data API being down) was found to compromise tutorial retrieval though, as a solution, fallback text answers were enabled.
- Language Issues: Presently the system works only on English-based inputs and multilingual inputs based on regional languages is envisaged in future versions.
- Generative Model Issues: While results from hybrid generation meant improved flexibility, over-reliance on generative potentialities could lead to some minor inaccuracies unless strong fine-tuning was enacted.

4.3 Input and Output format

Input.
What is the process for caste certificate?

Output.
Output of query, Information of the process regarding caste certificate. (see Fig. 4.)

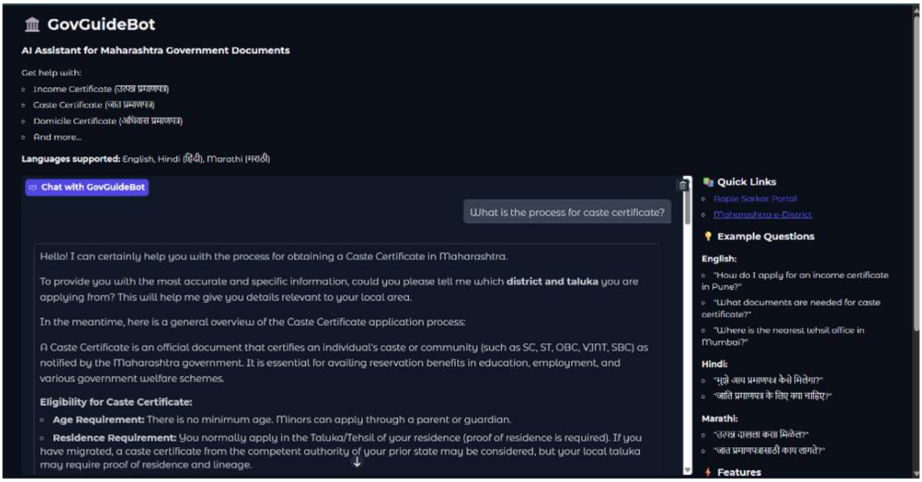

Fig. 4. Output of the query

4.4 Future Scope

In future we will be expanding our dataset and will be making our data increasing in recursive format, which will take input from user and will act as a self-learning model. This will enhance the user experience and give more accurate results as the query will be already present in the dataset of the self-trained model. GovGuideBot will scale mostly in the farmer schemes and expand its user area all across the country.

5 Conclusion

The paper presents an AI-enabled conversational agent, called GovGuideBot, to support the process of applying for government documentation on Income, Caste or Domicile Certification via one-on-one support based on location and process, natural language processing (NLP), and machine learning. GovGuideBot demonstrates awareness of the difficulties faced by citizens, particularly for example those in rural communities and with the lowest levels of digital literacies, attempting to access basic governments services. The evaluation findings show that GovGuideBot had a very high accuracy of intent recognition (92%), an average response time of 1.4 seconds across 651 different intents, and a 35% reduction in the number of mistakes across applications compared to self-guidance processes. In addition, anecdotal feedback from participants indicated that GovGuideBot: provided enhanced clarity.

REFERENCES

1. Musumeci, E., Brienza, M., Suriani, V., Nardi, D., Bloisi, D. D.: LLM-based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. In: Degen, H., Ntoa, S. (eds.) Artificial Intelligence in HCI, LNCS, pp. 98–117. Springer, Heidelberg (2024).
2. Iqbal, S., Pobee, S., Adhikari, A., Schooley, B.: MathBuddy: An LLM-based chatbot for elementary math education. In: Smith, B.K., Borge, M., Sottilare, R.A., Schwarz, J. (eds.) HCI International 2025 – Late Breaking Papers, HCII 2025, LNCS, vol. 16344, pp. 392–403. Springer, Cham (2026).
3. Mamalis, M.E., Kalampokis, E., Fitsilis, F., Theodorakopoulos, G., Tarabanis, K.: A large language model agent based legal assistant for governance applications. In: Janssen, M., Crompvoets, J., Gil-Garcia, J.R., Lee, H., Lindgren, I., Nikiforova, A., Viale Pereira, G. (eds.) Electronic Government, LNCS, pp. 286–301. Springer, Heidelberg (2024).
4. Tsai, J.C., Tsai, C.C., Tseng, S.P., Li, Y., Lin, H.M.: The evaluation of the LLM-based chatbot application on Chinese learning. In: Tseng, S.-P., Paul, A., Pan, J.-S., Favorskaya, M. (eds.) Advances in Intelligent Information Hiding and Multimedia Signal Processing, Volume 1, IIHMSP 2023, Smart Innovation, Systems and Technologies, vol. 415, pp. 293–300. Springer, Singapore (2025).
5. Arora, Y., et al.: Building legal intelligence: Designing and developing a chatbot with LLM. In: Bhateja, V., Hoong, A.L.S., Kong, J.D., Urooj, S. (eds.) Smart Computing Paradigms: Intelligence and Network Applications, SCI 2025, Lecture Notes in Networks and Systems, vol. 1682, pp. 1–12. Springer, Cham (2026).
6. Barbieri, L., Stroeh, K., Madeira, E.R.M., van der Aalst, W.M.P.: An LLM-based Q&A natural language interface to process mining. In: Delgado, A., Slaats, T. (eds.) Process Mining Workshops, LNBIP, pp. 5–17. Springer, Heidelberg (2025).
7. Cheng, Q., Jiao, X., Yang, M., Yang, M., Jiang, K., Yang, D.: Advancing autonomous driving safety through LLM enhanced trajectory prediction. In: Mastinu, G., Braghin, F., Cheli, F., Corno, M., Savaresi, S.M. (eds.) 16th International Symposium on Advanced Vehicle Control, Lecture Notes in Mechanical Engineering, pp. 496–502. Springer, Heidelberg (2024).
8. Coda-Giorgio, L., Fidone, G., Pollacci, L.: Multi-domain validation of LLM-based simulators via interpretable and latent representations. In: Džeroski, S., Levatić, J., Pio, G., Simidjievski, N. (eds.) Discovery Science, LNCS, pp. 411–426. Springer, Heidelberg (2025).
9. Angelopoulos, J., Manettas, C., Alexopoulos, K.: Industrial maintenance optimization based on the integration of large language models (LLM) and augmented reality (AR). In: Alexopoulos, K., Makris, S., Stavropoulos, P. (eds.) Advances in Artificial Intelligence in Manufacturing II, Lecture Notes in Mechanical Engineering, pp. 197–205. Springer, Heidelberg (2025).
10. Sam Prince Franklin, K., Thanish Reddy, D., Poonkodi, M., Woonna, V.A., Rachakonda, V.V.: Enhancing PDF information retrieval through a Gemini Pro LLM-powered chatbot. In: Sivakumar, P.D., Ramachandran, R., Pasupathi, C., Balakrishnan, P. (eds.) Computing Technologies for Sustainable Development, IRCCTSD 2024, Communications in Computer and Information Science, vol. 2362, pp. 355–370. Springer, Cham (2025).

11. Das, D., Rath, R.L., Singh, T., Mishra, S., Malik, V., Sobti, R., Brahma, B.: LLM-based custom Chatbot using LangChain. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds.) Innovative Computing and Communications, ICICC 2024, Lecture Notes in Networks and Systems, vol. 1020, pp. 257–267. Springer, Singapore (2024).
12. Young, R., Ghaharian, K., Golab, L., Kraus, S.W., Wells, S., Soligo, M.: Comparing LLM and human expert responses to problem gambling questions. In: Arabnia, H.R., Deligiannidis, L., Amirian, S., Ghareh Mohammadi, F., Shenavarmasouleh, F. (eds.) AI Revolution: Research, Ethics and Society, AIR-RES 2025, Communications in Computer and Information Science, vol. 2721, pp. 226–238. Springer, Cham (2026).
13. Zheng, J., Yilmaz, G., Han, J., Ahmed-Kristensen, S.: Digital transformation chatbot (DTchatbot): Integrating large language model-based chatbot in acquiring digital transformation needs. In: Nah, F.F.H., Siau, K.L. (eds.) HCI International 2025 – Late Breaking Papers, HCII 2025, Lecture Notes in Computer Science, vol. 16343, pp. 388–403. Springer, Cham (2026).
14. Saha, T.S., Mukherjee, J., Kar, M., Chakrabarti, A., Guha, K.: A LLM and explainable AI assistive model for improvement of user's query in a chatbot for women cancer awareness for Indian lay people.In: Goswami, S., Saha, S., Basu, K., Beed, R.S. (eds.) Data Management, Analytics and Innovation, ICDMAI 2025, Lecture Notes in Networks and Systems, vol. 1369, pp. 211–226. Springer, Singapore (2026).
15. Patle, A., Punjabi, M., Ajalkar, D., Bicholia, P.: Future trends and directions for knowledge graph embeddings based on visualization methodologies. In: Knowledge Graph-Based Methods for Automated Driving, pp. 159–175. Elsevier (2025).