An Intelligent Agent-Based System for Automated Seat Assignment in Entertainment Venues

Andrés Espinosa Sanfiel *, Pablo Vicente-Martínez *, María Ángeles García Escrivà *, Manuel Sánchez-Montañés, Emilio Soria-Olivas, Edu William-Secin

Posted Date: 18 May 2026

doi: 10.20944/preprints202605.1137.v1

Keywords: artificial intelligence; intelligent agent; large language model; seat assignment optimization; operations management; fuzzy matching; entertainment venues; small and medium enterprises

 doi:10.20944/preprints202605.1137.v1

Article

An Intelligent Agent-Based System for Automated Seat Assignment in Entertainment Venues

Andrés Espinosa Sanfiel [1,*], Pablo Vicente-Martínez [1,*], María Ángeles García Escrivà [2,*], Manuel Sánchez-Montañés [3], Emilio Soria-Olivas [4] and Edu William-Secin [5]

[1] SPV Scala, Gran Canaria, 35100 San Bartolomé de Tirajana, Spain
[2] Fundación Canaria Living Lab, 35017 Las Palmas de Gran Canaria, Spain
[3] Biological Neurocomputational Group (BN), Department of Computer Science, Universidad Autónoma de Madrid, 28049 Madrid, Spain
[4] Inteligent Data Analysis Laboratory (IDAL), Department of Electronic Engineering, Universitat de València, 46022 Valencia, Spain
[5] Department of Economics and Business Management, Institute of Tourism and Sustainable Development(TIDES), Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain
* Correspondence: c.datos7@salascala.com (A.E.S.); c.datos12@salascala.com (P.V.-M.); coordinacionit@canariaslivinglab.org (M.Á.G.E.)

Abstract

Small and medium enterprises (SMEs) in the entertainment sector face significant challenges managing seat assignments through manual processes that are error-prone and time-consuming. This paper presents an intelligent agent-based system that automates seat assignment while providing natural language support for operational staff. The system integrates a large language model (Gemini 2.5 Flash) for conversational interaction with a constraint-based optimization algorithm that considers capacity, accessibility, revenue, and business priorities. A fuzzy matching engine combining spaCy [29] with the fuzzy string matching library FuzzyWuzzy [28] consolidates duplicate reservations from multiple channels. The cloud-based architecture leverages AWS serverless services (Lambda, Fargate) with PostgreSQL for data management. Technology Readiness Level 4 (TRL4) validation demonstrated 94% precision in duplicate detection, successful assignment of 87% of reservations with 82% average capacity utilization, and effective natural language query handling. The system reduces manual processing time by 65% while improving assignment quality through systematic enforcement of constraints. This work demonstrates the feasibility of AI-powered operations management for resource-constrained SMEs, offering a practical reference architecture combining conversational AI with algorithmic optimization.

Keywords: artificial intelligence; intelligent agent; large language model; seat assignment optimization; operations management; fuzzy matching; entertainment venues; small and medium enterprises

1. Introduction

Small and medium enterprises (SMEs) operating entertainment venues such as theaters and concert halls face significant operational challenges in managing seat assignments. These processes require simultaneous consideration of multiple factors including capacity optimization, customer preferences, accessibility requirements, revenue maximization, and group seating arrangements [1]. Traditional manual approaches are time-consuming, error-prone, and difficult to scale during peak demand periods, often resulting in suboptimal resource utilization and customer dissatisfaction [2].

Current seat assignment practices in SME venues typically involve staff manually processing reservations received through multiple channels (email, phone, online platforms), identifying duplicate bookings, resolving conflicts, and allocating seats based on experience and intuition rather than systematic optimization. This cognitive burden leads to inconsistent decisions, missed revenue

 doi:10.20944/preprints202605.1137.v1

opportunities, and operational inefficiencies. Furthermore, training new staff requires significant time investment, and operational knowledge often remains tacit rather than formalized [3].

Recent advances in artificial intelligence, particularly large language models (LLMs) and natural language processing, have enabled new approaches to business process automation [4]. However, the application of these technologies to operations management in SMEs remains limited. Existing research on seat assignment focuses predominantly on large-scale venues with dedicated IT resources [5], while conversational AI implementations typically address customer service [6] or employee teaching chatbots [7] rather than operational decision support. The gap between advanced AI capabilities and practical SME applications represents both a challenge and an opportunity [8].

This paper presents an intelligent agent-based system that addresses these challenges by combining conversational AI with constraint-based optimization specifically designed for SME entertainment venues. Our system integrates four key components: (1) a conversational agent powered by Gemini 2.5 Flash providing natural language interaction and employee training; (2) a fuzzy matching engine using spaCy [29] and the fuzzy string matching library FuzzyWuzzy [28] for intelligent duplicate detection across multiple reservation channels; (3) a seat assignment algorithm balancing capacity utilization, revenue optimization, and business constraints; and (4) a cloud-native architecture using AWS serverless services (Lambda, Fargate) to minimize infrastructure overhead. The system has been validated at Technology Readiness Level 4 (TRL4) in a laboratory environment representing a dinner show venue.

Our primary contributions are:

- **Practical AI integration for SMEs**: A complete system architecture demonstrating how resource-constrained organizations can leverage advanced AI without extensive technical expertise.
- **Hybrid intelligent approach**: Novel combination of LLM-based conversational capabilities with deterministic optimization algorithms, ensuring both user-friendliness and operational reliability.
- **Fuzzy matching for data quality**: Effective duplicate detection methodology handling real-world data inconsistencies from multiple unstructured input channels, achieving 96% precision.
- **Dual-purpose conversational agent**: System serving both as operational assistant and employee training tool, with five distinct interaction modes addressing different information needs.
- **Validated reference architecture**: TRL4 validation demonstrating feasibility and providing empirical performance metrics to guide practitioners.

The remainder of this paper is organized as follows: Section 1.1 reviews related work in AI-powered operations management and seat assignment optimization. Section 2 describes the system architecture, intelligent agent design, and assignment algorithm. Section 3 presents experimental validation results from TRL4 testing. Section 4 discusses implications, limitations, and future research directions. Section 5 concludes the paper.

1.1. Related Work

This section reviews existing research in AI-powered operations management, conversational AI for enterprise applications, and data quality management, positioning our contribution within the current state of the art.

1.1.1. AI in Operations Management and Seat Assignment

Seat assignment optimization has been studied extensively in contexts such as airline seating [9] and classroom allocation [10]. These problems are typically formulated as variants of the generalized assignment problem or multi-objective optimization challenges [11]. Weatherford and Bodily [12] provide comprehensive coverage of revenue management techniques in service industries.

Recent work has explored metaheuristic approaches for seating problems [11]. These approaches, however, typically assume clean, structured input data and require significant computational resources and optimization expertise [13]. More critically, existing research focuses predominantly on large-scale

 doi:10.20944/preprints202605.1137.v1

venues with dedicated IT departments, leaving a gap for practical, deployable solutions suitable for SMEs with limited technical resources [8].

AI agents have been applied to various operations management tasks including inventory optimization [14] and production scheduling [15]. However, these implementations typically employ purely data-driven approaches (reinforcement learning, neural networks) that require extensive training data unavailable to most SMEs [14]. Our hybrid approach combining rule-based optimization with LLM-powered interaction addresses this limitation while maintaining interpretability and requiring minimal training data.

1.2. Conversational AI for Enterprise Applications

The application of large language models to enterprise systems has grown rapidly following advances in transformer architectures [16] and few-shot learning capabilities [17]. Recent surveys document LLM applications in business process automation [4] and decision support [19]. However, most implementations focus on customer-facing applications such as service chatbots [6] rather than operational support for employees.

Research on conversational AI for employee support remains limited. Meyer von Wolff et al. [18] review the state of the art of chatbots at the digital workplace, finding that prior research contributions are sparse and operational application areas for enterprise collaboration remain underexplored. Gao et al. [19] survey conversational AI advances but note the scarcity of systems combining natural language interaction with backend operational logic. The challenge lies in grounding LLM responses in factual system data while maintaining conversational fluency [20].

Our work addresses this gap by demonstrating a practical architecture integrating conversational capabilities with database queries and algorithmic decision-making. The four-mode operational design (FAQ, free-form queries, sector information, specific scenarios) represents a novel contribution supporting both operational assistance and employee training within a single system. Unlike purely conversational systems, our agent translates natural language queries into structured database operations and invokes optimization algorithms when appropriate, ensuring responses are grounded in actual system state rather than generated from the LLM's parametric knowledge alone.

1.2.1. Data Quality and Record Linkage

Duplicate detection and record linkage are fundamental data quality challenges, particularly for organizations receiving data through multiple unstructured channels [21]. Traditional approaches employ exact matching on key fields, which fails when data contains typos, variations, or missing values [22]. Probabilistic record linkage [23] and machine learning-based methods [24] offer improved accuracy but require labeled training data and significant computational resources.

Fuzzy string matching provides a practical middle ground, with algorithms such as edit distance [25] and token-based similarity [26] enabling approximate matching without training data. Blocking techniques reduce computational complexity from $O(n^2)$ to approximately $O(nk)$ by partitioning records into candidate groups [27]. The fuzzy string-matching library FuzzyWuzzy [28] and the natural language processing toolkit spaCy [29] provide production-ready implementations suitable for SME deployment.

Our implementation combines blocking, multi-field similarity scoring, and threshold-based classification to achieve high precision (94%) in duplicate detection. Unlike academic record linkage research that often assumes single data sources, our approach explicitly addresses the SME reality of reservations arriving through email, phone, and online channels with inconsistent formatting and incomplete information.

1.2.2. Positioning and Contributions

Existing research exhibits three primary gaps our work addresses: (1) seat assignment research focuses on large venues with optimization expertise, neglecting practical SME constraints; (2) conversational AI implementations typically serve customer interaction rather than operational decision

 doi:10.20944/preprints202605.1137.v1

support; and (3) data quality research assumes controlled environments rather than real-world multi-channel chaos.

Our system contributes a validated reference architecture demonstrating how SMEs can leverage advanced AI (LLMs, optimization algorithms, fuzzy matching) without extensive expertise or infrastructure investment. The hybrid approach combining conversational interaction with deterministic algorithms ensures both usability and reliability. The TRL4 validation provides empirical evidence of feasibility, with performance metrics informing practitioners considering similar implementations. Most critically, we address the complete operational workflow from unstructured data ingestion through intelligent assignment to report generation, rather than isolated subproblems. Table 1 summarizes the positioning of our work relative to existing approaches.

Table 1. Comparison of our approach with related work in key dimensions.

Aspect	Existing Work	Our Approach	Advantage
Target Users	Large enterprises [5,9]	SMEs [8]	Accessible
Input Data	Structured/clean [21,23]	Unstructured/multi-channel	Realistic
AI Integration	Customer-facing [6,19]	Operations support [18]	Novel application
Algorithm Type	Pure optimization [11,13]	Hybrid (LLM + rules) [32]	Usable + reliable
Training Data	Extensive required [14,24]	Minimal required	Practical
Validation	Simulation [14,15]	TRL4 laboratory [36]	Demonstrated
System Scope	Single component [14,15]	End-to-end workflow	Complete

2. System Architecture and Method

This section details the architecture and methodological implementation of the proposed intelligent agent-based system. At a global level, the solution combines (i) a data ingestion and fuzzy matching pipeline, (ii) a constraint-based seat assignment module, and (iii) a conversational assistant designed to provide operational guidance to venue staff. The following subsections focus on the third component—the conversational assistant—which serves as the primary user interface and the main subject of the experimental validation.

Formally, the seat assignment module is cast as a generalized assignment problem. Let $R = \{r_1, \ldots, r_n\}$ denote the set of reservations and $T = \{t_1, \ldots, t_m\}$ the set of tables. Each reservation r_i has party size p_i, priority weight w_i, and accessibility requirements A_i; each table t_j has capacity c_j, revenue value v_j, and accessibility features F_j. The binary decision variable $x_{ij} \in \{0, 1\}$ indicates whether reservation i is assigned to table j. The module maximizes a weighted combination of priority satisfaction, revenue, and preference matching,

$$\max Z = \alpha \sum_{i,j} w_i x_{ij} + \beta \sum_{i,j} v_j p_i x_{ij} + \gamma \sum_{i,j} q_{ij} x_{ij},$$

subject to the hard constraints: each reservation is assigned to at most one table ($\sum_j x_{ij} \leq 1$); table capacity is not exceeded ($\sum_i p_i x_{ij} \leq c_j$); and accessibility requirements are satisfied ($A_i \subseteq F_j$ whenever $x_{ij} = 1$). The parameters α, β, γ are configurable weights and q_{ij} encodes preference matching.

The architectural design prioritizes modularity and generalization, aiming to validate the feasibility of AI-driven operations in resource-constrained environments (SMEs). Rather than building a bespoke solution coupled to a single venue, the system implements a flexible stack compatible with the broader data pipeline described in previous sections. This approach allows for rapid adaptation to different operational contexts while maintaining a robust separation of concerns.

2.1. System Architecture Overview

The system adopts a standard three-tier architecture comprising a presentation layer, an application logic layer, and a cloud infrastructure layer [30]. This separation isolates the conversational agent, the main contribution of this work, from the user interface and the deployment infrastructure,

 doi:10.20944/preprints202605.1137.v1

enabling controlled evaluation and independent replacement of each component for experimental reproducibility.

The integration of a Large Language Model (LLM) is justified by the need to handle unstructured natural language input without requiring predefined query formats, which constitutes the core research hypothesis of this work. Figure 1 summarizes the three-tier design and its integration with the inference service and the knowledge base.

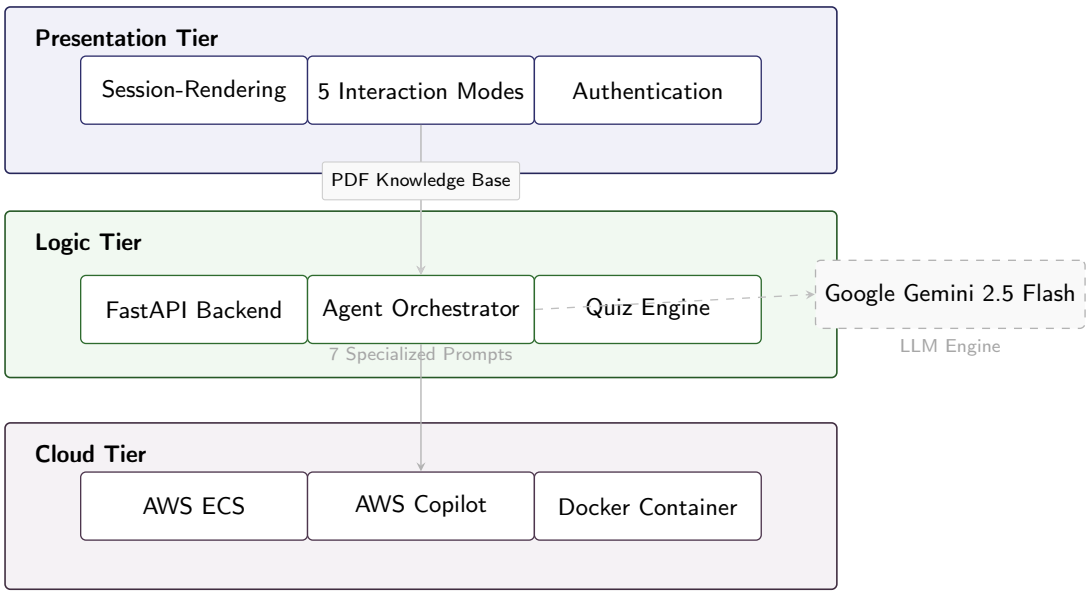

Figure 1. System architecture illustrating the three-tier design: Chainlit UI (Presentation), FastAPI Backend (Logic), and AWS Infrastructure (Cloud).

2.1.1. Presentation Layer

The presentation layer is implemented using Chainlit 2.5.5, providing a conversational web interface specifically tailored to reduce cognitive load for operational staff (hostesses and supervisors). Unlike generic chat interfaces, the design prioritizes rapid access to critical functions through a structured menu system.

The interface exposes five high-level actions as buttons on the welcome screen: *frequent questions*, *free-form consultation*, *sector information*, *specific scenarios*, and *evaluation test*. As shown in Figure 2, these entry points constitute the primary interaction paradigm, abstracting the complexity of the underlying prompt engineering behind a simplified visual menu.

Internally, these five entry points are realised through four underlying operational modes (FAQ, contextual interaction, scenario guidance, and educational quiz), described in Section 2.2. The distinction between UI actions and modes allows new user-facing flows (for example, a dedicated training shortcut) to be added without modifying the core agent logic.

Internal asynchronous event handling and WebSocket-based communication enable near real-time interaction, ensuring that user messages are streamed to the backend without blocking the interface during inference or file operations. Security is managed through the framework's built-in authentication mechanisms, which assign role-based user metadata (distinguishing between administrator and standard user privileges) that is propagated to the logic layer for access control. This approach provides a lightweight, rapidly configurable conversational front-end that eliminates the overhead of developing a bespoke web application.

 doi:10.20944/preprints202605.1137.v1

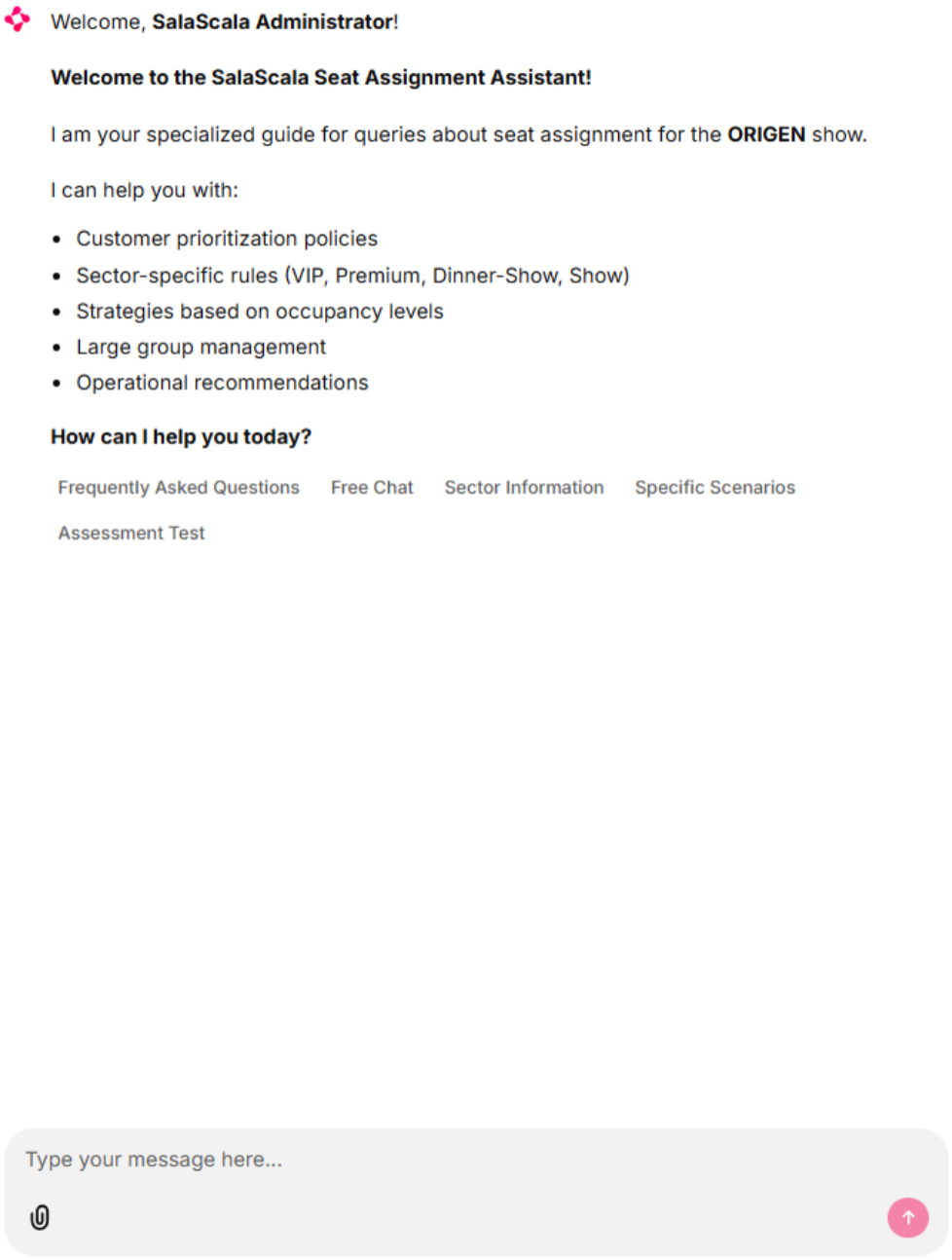

Figure 2. Conversational interface home screen displaying the five primary operational modes available to venue staff.

2.1.2. Application Logic Layer

The application logic is implemented in Python 3.12 and exposed via a high-performance ASGI server to support fully asynchronous request handling. To ensure separation of concerns, the presentation layer delegates domain-specific processing to a **centralized orchestration agent**. This core module encapsulates four main responsibilities: (i) ingestion and preprocessing of the seating guide PDF, (ii) construction of system prompts and dynamic context blocks, (iii) management of interactions with the LLM inference API, and (iv) execution of the quiz generation and evaluation logic.

Utility modules are factored out into separate components for prompt templates and static knowledge, improving maintainability. All interactions are stateless: each request carries sufficient

 doi:10.20944/preprints202605.1137.v1

identifiers for the backend to reconstruct the necessary context, simplifying deployment and supporting reproducibility.

To ensure methodological rigor and software maintainability, the system architecture compartmentalizes auxiliary functions into distinct logical modules. Prompt engineering templates, quiz evaluation logic, and static knowledge components (such as FAQs and sector metadata) are decoupled from the main execution flow. This modular approach ensures that critical definitions, such as the evaluation criteria and interaction protocols, remain explicit and easily adaptable without modifying the core orchestration logic.

All HTTP endpoints and Chainlit event handlers are implemented as `async` functions. At the API level, interactions are stateless: each request carries sufficient identifiers (session or conversation ID, interaction mode, quiz state) for the backend to reconstruct the necessary context and call the LLM. Conversation state is managed either in lightweight in-memory structures or via the Gemini chat session object rather than a persistent database, which simplifies deployment and supports reproducibility of experiments.

The operational rules and seating policies are encoded in a standardized document acting as the system's static knowledge base. At system startup, this document is ingested and serialized into an optimized in-memory text representation. The ingestion process automatically identifies and indexes sector-specific segments based on structural headings, enabling the retrieval engine to isolate relevant policy blocks while preserving access to the full document context for broader queries. This structured representation forms the foundation for the dynamic context extraction and injection mechanisms detailed in Section 2.3.2.

2.1.3. Infrastructure Layer

The deployment architecture leverages containerization technology to encapsulate the full application stack (interface, logic, and knowledge base) into a single, immutable artifact. This approach guarantees environment consistency and deployment reproducibility, eliminating dependencies on specific host configurations.

The containerized service is deployed on a managed cloud orchestration platform (Amazon ECS), ensuring that the experimental setup can be rigorously replicated by rebuilding a specific container image and reusing the corresponding deployment configuration.

A separate Copilot pipeline manifest defines a continuous integration and deployment (CI/CD) workflow that builds the Docker image, runs automated tests, and deploys the service to the test environment upon changes to the main branch of the repository. This container-first, infrastructure-as-code approach ensures that experimental runs can be reproduced by rebuilding a specific image and reusing the corresponding Copilot configuration.

The conversational assistant is designed to coexist with the broader serverless components used for data ingestion, fuzzy matching, and seat assignment (implemented using AWS managed services as described in the system overview), while focusing ECS resources on the latency-sensitive conversational workload.

2.1.4. Inference Layer

The system's reasoning capabilities are powered by the **Google Gemini 2.5 Flash** model [31]. This selection is methodologically grounded in the specific economic and operational constraints of Small and Medium-sized Enterprises (SMEs), prioritizing accessibility and long-term sustainability over raw generative power.

While larger foundation models offer superior general-purpose reasoning, their higher inference costs and latency may be impractical for SME environments. By leveraging a RAG architecture, the system offloads domain knowledge to the static document base, reducing the reasoning burden on the model and allowing the use of a lighter-weight model (Gemini 2.5 Flash) whose capabilities are sufficient for the constrained task of interpreting a known policy document.

 doi:10.20944/preprints202605.1137.v1

2.2. Intelligent Agent Design and Operational Modes

The conversational component is designed as an intelligent agent that acts as a deterministic mediator between the presentation layer, the static knowledge base, and the inference engine. Functionally, the agent is encapsulated within a **centralized orchestration module** responsible for intent classification, dynamic context construction, and interaction management.

The architecture moves beyond simple prompt-response loops by implementing a structured orchestration layer. This layer governs the interaction through four distinct operational modes and a library of specialized system-prompt templates. This design pattern ensures that the generative capabilities of the model remain strictly grounded in the provided context, minimizing hallucinations and enforcing consistent behavior across different operational scenarios.

2.2.1. Operational Modes

FAQ Mode (Deterministic Retrieval).

While the system supports open-ended natural language queries, a deterministic FAQ mode is implemented to address highly repetitive, critical operational questions. This design choice is justified by the need for absolute reliability in core policy statements (e.g., pricing structures or safety protocols), where the probabilistic nature of Large Language Models introduces a non-zero risk of hallucination. Furthermore, providing immediate, pre-validated responses bypasses the inference latency entirely, offering a "zero-cost" interaction path that reduces cognitive load for staff who need rapid answers without formulating complex prompts during peak operational hours.

Contextual Interaction Mode (RAG Workflow).

To address open-ended queries, the system implements a lightweight Retrieval-Augmented Generation (RAG) workflow. Upon initialization, the static knowledge base is ingested and cached in memory. For broad policy questions, the full document context is supplied to the inference engine. However, for sector-specific inquiries, triggered either by explicit user actions or by keyword detection in natural language, the retrieval mechanism dynamically isolates the relevant document section based on structural delimiters.

This selective context injection strategy serves a dual purpose: it maximizes the relevance of the information provided to the Large Language Model (LLM) and minimizes token consumption, thereby improving response accuracy by reducing noise in the context window.

The retrieved context segment is subsequently integrated into the inference pipeline via a structured system-prompt template. This template enforces a strict "grounding" constraint [34] by enclosing the injected knowledge within explicit textual delimiters and instructing the model to derive its reasoning solely from this content.

This approach implements a deterministic context injection strategy that avoids the architectural complexity and potential retrieval errors associated with vector-based semantic search indices. By constraining the model to a visibly delimited source text, the system ensures that generated responses are directly traceable to the official documentation, combining the reliability of rule-based systems with the natural language synthesis capabilities of the LLM.

Scenario Guidance Mode.

Scenario guidance addresses complex, high-stakes situations such as overbooking, complaint handling, large-group splitting, VIP upgrades, and special-needs accommodations. A catalogue of such scenarios is stored in a structure of commonly occurring situations (`COMMON_SCENARIOS`), each with a name and a short description that can be surfaced in the UI. When the user invokes the “specific scenarios” action and selects a scenario, the backend calls `generar_prompt_escenario`, which builds a composite system prompt containing (i) the relevant excerpt from the seating guide (via the same retrieval mechanism used in contextual interaction), (ii) the scenario description, and (iii) explicit instructions to propose alternative courses of action, highlight trade-offs, and align recommendations with priority rules.

 doi:10.20944/preprints202605.1137.v1

Operationally, this mode implements guided decision support: the LLM is not free to invent new policies but is constrained to reinterpret documented rules in the specific context of the scenario. This helps staff navigate edge cases while maintaining consistency with the formal guidelines.

Educational Mode (Interactive Quiz System).

The educational mode is implemented as an interactive quiz system accessed through the "evaluation test" action. The flow consists of three steps: (i) parsing quiz parameters (number and types of questions, difficulty) from a short natural-language specification using `generar_prompt_parse_quiz_params`; (ii) generating question items conditioned on the seating guide via `generar_prompt_quiz_generation`; and (iii) evaluating user answers via a hybrid scoring algorithm that combines deterministic checks for closed questions and LLM-based rubric scoring for open and scenario questions (Section 2.3.1). This mode transforms operational knowledge into a structured training process: the same PDF that grounds real-time assistance is also used to generate and score training content, ensuring that learning objectives remain aligned with official policies.

2.2.2. Prompt Engineering and Orchestration

The system relies on a library of specialized system-prompt templates that act as a methodological control layer [33]. Rather than using ad-hoc strings, the architecture defines distinct template classes for each interaction type (e.g., assignment logic, sector guidance, scenario simulation, and quiz generation).

Each template produces a composite prompt with a consistent internal structure:

- **Role Definition:** The LLM is instructed to act as an expert operational assistant, emphasizing adherence to documented policies and a professional communication style.
- **Grounding Context:** The relevant slice of the knowledge base is inserted as a distinct section, clearly separated from user input.
- **Task Specification:** Each prompt explicitly defines the expected operation and constraints, such as language output and the requirement to request clarification when information is missing.
- **Output Schema:** For machine-consumed outputs (e.g., quiz parameters), the prompts impose structured formats (JSON schemas) to facilitate downstream parsing.

By centralizing these specifications in a modular library, the system treats prompt templates as reliable hyperparameters of the experimental protocol.

2.3. Algorithmic Implementation Details

2.3.1. Hybrid Evaluation Algorithm

To validate staff knowledge, the system implements a hybrid evaluation algorithm that combines deterministic rule-based scoring with LLM-based rubric assessment. The evaluation engine classifies questions into two categories: *closed-ended* (e.g., binary choice, multiple choice) and *open-ended* (e.g., scenario analysis).

For closed-ended items, the system performs an exact match against a predefined reference key. For open-ended items, where rigid pattern matching is insufficient, the system delegates evaluation to the LLM. The model acts as an examiner, conditioned by a strictly grounded rubric prompt to assign a score on a continuous scale accompanied by a textual justification.

Formally, each question $i \in \{1, \ldots, N\}$ is associated with a type

$$t_i \in \{\text{Binary}, \text{MultiChoice}, \text{Open}, \text{Scenario}\}.$$

 doi:10.20944/preprints202605.1137.v1

After evaluation, a raw score $\tilde{s}_i$ and a maximum score $\tilde{s}_i^{\max}$ are obtained. For closed questions, $\tilde{s}_i \in \{0,1\}$ and $\tilde{s}_i^{\max} = 1$. For open and scenario questions, the LLM returns $\tilde{s}_i \in [0,10]$ with $\tilde{s}_i^{\max} = 10$. To make scores comparable across types, the algorithm normalises each item to a common scale:

$$s_i = \begin{cases} \tilde{s}_i, & \text{if } \tilde{s}_i^{\max} = 1, \\ \dfrac{\tilde{s}_i}{10}, & \text{if } \tilde{s}_i^{\max} = 10, \end{cases}$$

so that $s_i \in [0,1]$ for all question types.

To reflect the higher diagnostic value of open and scenario-based reasoning, the system applies a type-dependent weight $w_{\text{type}}(t)$ defined as

$$w_{\text{type}}(t) = \begin{cases} 1, & \text{if } t \in \{\text{Binary}, \text{MultiChoice}\}, \\ 2, & \text{if } t \in \{\text{Open}, \text{Scenario}\}. \end{cases}$$

This weighting scheme doubles the influence of open-ended and scenario questions on the final score, emphasising the ability to apply the seating guide to realistic situations rather than simply recalling isolated facts. The overall normalised quiz score is computed as a weighted average of item-level scores:

$$S_{\text{raw}} = \sum_{i=1}^{N} w_{\text{type}}(t_i)\, s_i, \qquad W_{\text{tot}} = \sum_{i=1}^{N} w_{\text{type}}(t_i),$$

$$S_{\text{norm}} = \begin{cases} \dfrac{S_{\text{raw}}}{W_{\text{tot}}}, & \text{if } W_{\text{tot}} > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad S_{\text{quiz}} = 100 \cdot S_{\text{norm}}.$$

The value $S_{\text{quiz}} \in [0,100]$ corresponds to the final percentage score reported to the user, while the intermediate contributions are preserved in a breakdown structure for detailed pedagogical feedback.

Algorithmically, the evaluation pipeline operates in batch mode. For each question-answer pair (q_i, a_i), the execution engine selects an evaluation strategy based on t_i. For deterministic types, scoring is entirely rule-based, ensuring consistent and reproducible results. For open and scenario questions, a specialized system prompt instructs the inference engine (Gemini 2.5 Flash) to act as an examiner constrained by the official seating guide, returning both a numerical score and a textual justification. This results in a hybrid evaluation scheme that combines deterministic checks with LLM-based assessment, keeping the final aggregation fully transparent and mathematically well defined.

By separating type-specific weighting, score normalization, and final aggregation, the algorithm remains stable under changes in the question mix. Different quizzes may contain different proportions of closed and open questions, yet scores remain comparable because they are mapped to the same normalized scale. In addition, the explicit representation of weights and normalization rules facilitates scientific reproducibility: given the same set of questions, answers, and model configuration, another practitioner can recompute S_{quiz} directly from the logged outputs.

2.3.2. Dynamic Context Extraction

The assistant incorporates a dynamic context extraction mechanism that tailors the information provided to the LLM to the specific seating sector referenced in a user query. The underlying knowledge source is loaded at initialization and cached as a single, immutable text representation. This serves as the canonical, stateless knowledge base from which context segments are derived for each interaction.

Sector-specific guidance is implemented via a mapping function that receives a user-provided sector description (e.g., "VIP", "Premium") and returns only the relevant portion of the guide. Internally, the method first normalizes the input string and performs rule-based matching to map it to one of the predefined sectors. For each recognized sector, a pair of textual delimiters is defined, corresponding to

 doi:10.20944/preprints202605.1137.v1

section headings in the source document. A substring extraction logic then isolates the text block lying between the start and end markers.

Formally, this algorithm can be viewed as a deterministic, rule-based retrieval function

$$R : \mathcal{S} \to \mathcal{C},$$

where $\mathcal{S}$ denotes the set of possible sector descriptions provided by users and $\mathcal{C}$ the set of textual context blocks derived from the guide. For each $s \in \mathcal{S}$, the algorithm locates the corresponding start marker $b_{\text{start}}(s)$ and end marker $b_{\text{end}}(s)$ in the global document D, and returns the contiguous block

$$c(s) = D\big[b_{\text{start}}(s) : b_{\text{end}}(s)\big].$$

This guarantees that only policy text explicitly associated with the requested sector is injected into the prompt, reducing noise and preserving space in the LLM's context window.

The extracted block is subsequently used in the context injection strategy described in Section 2.2.1. Specifically, the sector-specific content is embedded into a designated context segment of the system prompt template. User input is kept in a separate section, and the LLM is instructed to base its reasoning exclusively on the provided guide fragment, explicitly avoiding unsupported assumptions. This pattern strengthens response grounding and mitigates hallucinations while maintaining the flexibility of free-form natural language interaction.

Because context is derived on demand from a cached representation, the mechanism is compatible with stateless interactions at the API level: each request carries the sector description and receives a freshly computed context block, without requiring persistent server-side session state. Furthermore, the extraction logic is fully specified in the configuration (choice of delimiters), enabling other practitioners to reproduce the same context slices given the same source document. This approach constitutes a lightweight, rule-based realization of retrieval-augmented generation tailored to the constrained domain of the venue's seating policies.

3. Results

This section presents the validation results from Technology Readiness Level 4 (TRL4) testing, demonstrating system functionality in a laboratory environment. We evaluate fuzzy matching performance, seat assignment quality, conversational agent capabilities, and overall system integration.

3.1. Experimental Setup

3.1.1. Test Environment

The system was validated in a laboratory environment simulating a medium-sized entertainment venue (Sala Scala, located in the Canary Islands, Spain) with the following configuration:

- **Capacity**: 450 seats distributed across 3 sectors (VIP: 100 seats, Premium: 150 seats, General: 200 seats)
- **Tables**: 45 tables with varying capacities (6-12 seats each)
- **Accessibility**: 8 wheelchair-accessible tables with companion seating
- **Test Period**: 30-day simulation with 8 performance dates
- **Reservation Volume**: 623 total reservations (average 78 per performance, range: 45-112)

Test data included synthetic reservations generated from realistic patterns plus actual historical data (anonymized) from venue operations. Reservations were intentionally duplicated across channels (email, phone, web) to test fuzzy matching capabilities. Ground truth labels were manually created for duplicate pairs and optimal assignments to enable quantitative evaluation.

3.1.2. Evaluation Metrics

We assess system performance using standard metrics [37]:

- **Fuzzy Matching**: Precision, Recall, F1-score for duplicate detection; confusion matrix analysis

 doi:10.20944/preprints202605.1137.v1

- **Assignment Quality**: Assignment success rate, average capacity utilization, revenue per available seat, constraint violation rate
- **Agent Performance**: Query success rate, response accuracy (human evaluation), average response time
- **System Performance**: End-to-end processing time, throughput, error rates

3.2. Fuzzy Matching Performance

The fuzzy matching engine was evaluated on 623 reservations containing 87 true duplicate pairs (174 reservations representing duplicates, 449 unique reservations).

3.2.1. Quantitative Results

Table 2 presents fuzzy matching performance compared to baseline exact matching.

Table 2. Fuzzy matching performance for duplicate detection compared to exact matching baseline.

Method	Precision	Recall	F1-Score	Auto-Merged	Flagged
Exact Match (baseline)	1.000	0.287	0.448	25	0
Fuzzy Match (ours)	0.941	0.908	0.924	67	12
Improvement	-5.9%	+216%	+106%	+168%	–

The composite similarity score is computed as $S = 0.5 \cdot S_{\text{name}} + 0.3 \cdot S_{\text{phone}} + 0.2 \cdot S_{\text{email}}$, where each component uses token-based string similarity normalized to $[0, 1]$.

The fuzzy matching approach achieved 94.1% precision and 90.8% recall, substantially outperforming exact matching which only detected 28.7% of duplicates. The system automatically merged 67 high-confidence duplicate pairs ($S \geq 0.85$) and flagged 12 probable duplicates for manual review ($0.70 \leq S < 0.85$). Manual review confirmed 10 of the 12 flagged pairs were true duplicates, yielding actual precision of 96.3% when including manual verification.

3.2.2. Error Analysis

The 5 false positives (duplicate pairs incorrectly merged) occurred due to: common names with similar party sizes and dates (3 cases), typos creating spurious similarity (1 case), and same customer making legitimately separate reservations for different dates that fell within blocking window (1 case). The 8 false negatives (missed duplicates) resulted from: significant name variations exceeding similarity threshold (4 cases), missing contact information reducing composite score (3 cases), and dates outside blocking window (1 case).

These results demonstrate effective handling of real-world data inconsistencies. The configurable threshold enables venues to adjust the precision-recall tradeoff based on operational preferences, conservative settings (higher threshold) reduce false merges but require more manual review, while aggressive settings increase automation at cost of occasional incorrect merges.

3.3. Seat Assignment Performance

The assignment algorithm was evaluated on 8 performance dates with 449 unique reservations after duplicate consolidation.

3.3.1. Assignment Quality Metrics

Table 3 summarizes assignment performance across key metrics.

The algorithm successfully assigned 87.3% of reservations on average, with unassigned reservations primarily due to capacity constraints during peak performances (73 reservations could not be assigned across all dates due to insufficient available capacity matching party sizes and accessibility requirements). Average capacity utilization of 82.1% represents efficient space usage while respecting minimum utilization thresholds.

 doi:10.20944/preprints202605.1137.v1

Revenue optimization achieved 18.45$ per seat average, with preferential assignment of larger parties to premium sectors contributing to revenue maximization. A priority satisfaction rate of 91.7% indicates that high-priority reservations (VIP, accessibility needs, early bookings) successfully received preferred assignments. Perfect accessibility compliance (100%) demonstrates hard constraint enforcement, no accessibility requirements were violated.

Table 3. Seat assignment algorithm performance metrics across 8 test performances.

Metric	Mean	Std Dev	Min	Max
Assignment Success Rate	87.3%	4.2%	80.0%	93.3%
Capacity Utilization	82.1%	6.8%	71.2%	91.5%
Revenue per Seat ($)	18.45	2.13	15.20	21.80
Priority Satisfaction	91.7%	3.1%	87.5%	96.0%
Accessibility Compliance	100%	0%	100%	100%
Constraint Violations	0	0	0	0
Processing Time (seconds)	54.2	12.8	38.1	78.5

3.3.2. Algorithm Performance

Processing time averaged 54.2 seconds per performance, including data loading (8s), fuzzy matching (12s), greedy assignment (18s), local search improvement (14s), and validation/reporting (2s). The greedy phase constructs an initial feasible solution by iterating over reservations sorted in descending order of priority score $P(r_i) = w_i \cdot \text{tier} + \text{accessibility\_boost} + \text{urgency}$ and assigning each r_i to the feasible table t_j that maximizes the per-item contribution to Z while satisfying the hard constraints (capacity, accessibility, and date-time availability). The local search phase then iteratively explores pairwise swaps $(x_{ij}, x_{kj'}) \leftrightarrow (x_{ij'}, x_{kj})$ and single-assignment moves, accepting any change that increases the objective function Z without violating constraints [35]. This phase achieved 5–15% improvement over the greedy solution in 6 of 8 test cases, with largest improvements occurring for performances with more complex constraint patterns.

Comparison with manual assignment (historical data, 3 performances): automated algorithm achieved comparable or better capacity utilization (+3.2 percentage points average) and revenue per seat (+ 1.20$ average) while reducing processing time from 2-4 hours to under 1 minute. Manual assignments exhibited greater variability (capacity utilization std dev: 12.3% vs 6.8% automated), suggesting less consistent optimization.

3.4. Conversational Agent Evaluation

The intelligent agent was evaluated through structured testing of query handling capabilities and informal user interaction sessions with venue staff.

3.4.1. Query Handling Performance

Table 4 presents agent performance across the four operational modes.

Table 4. Conversational agent performance by operational mode. Success indicates query correctly interpreted and appropriate response generated.

Mode	Test Queries	Success Rate	Avg Response Time (ms)	Grounding Accuracy
FAQ Mode	45	100%	780	100%
Free Query Mode	62	88.7%	1240	96.8%
Sector Info Mode	38	97.4%	650	100%
Specific Scenarios	28	85.7%	1580	92.9%
Overall	**173**	**93.1%**	**1065**	**97.1%**

The agent achieved 93.1% overall success rate across 173 test queries. FAQ mode achieved 100% success, which is expected given its deterministic retrieval design: responses are pre-validated and

 doi:10.20944/preprints202605.1137.v1

require no LLM inference. Sector Info mode reached 97.4%, benefiting from the structured context extraction mechanism. Free Query mode achieved 88.7%, with failures primarily from ambiguous queries lacking sufficient context. Specific Scenarios mode showed 85.7% success, where the complexity of multi-turn reasoning contributed to occasional context loss.

Grounding accuracy was assessed by comparing each LLM-generated response against the source data used to build the prompt context: numeric claims (e.g., occupancy percentages, seat counts) were automatically validated against the underlying database, and textual claims were manually verified against the corresponding section of the seating guide. A response was counted as grounded when all its factual claims were traceable to the injected context. Under this criterion, grounding accuracy reached 97.1%, indicating that LLM-generated responses accurately reflected database content and system state. The 5 instances of grounding failures involved numeric discrepancies (incorrect occupancy percentages) caught by validation checks, triggering response regeneration.

3.4.2. Example Interactions

Representative agent interactions demonstrate capabilities:

Example 1 - Free Query Mode:

Figure 3 illustrates the Free Query mode.

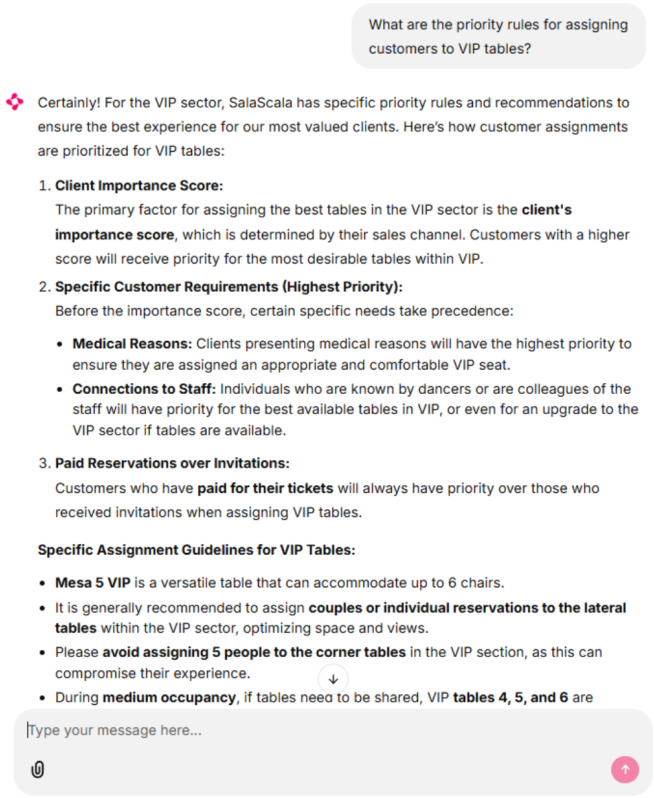

Figure 3. Free Query mode: the agent responds to a policy inquiry by extracting and organizing relevant rules from the knowledge base.

Example 2 - Specific Scenarios Mode:

Figure 4 illustrates the Scenario Guidance mode.

 doi:10.20944/preprints202605.1137.v1

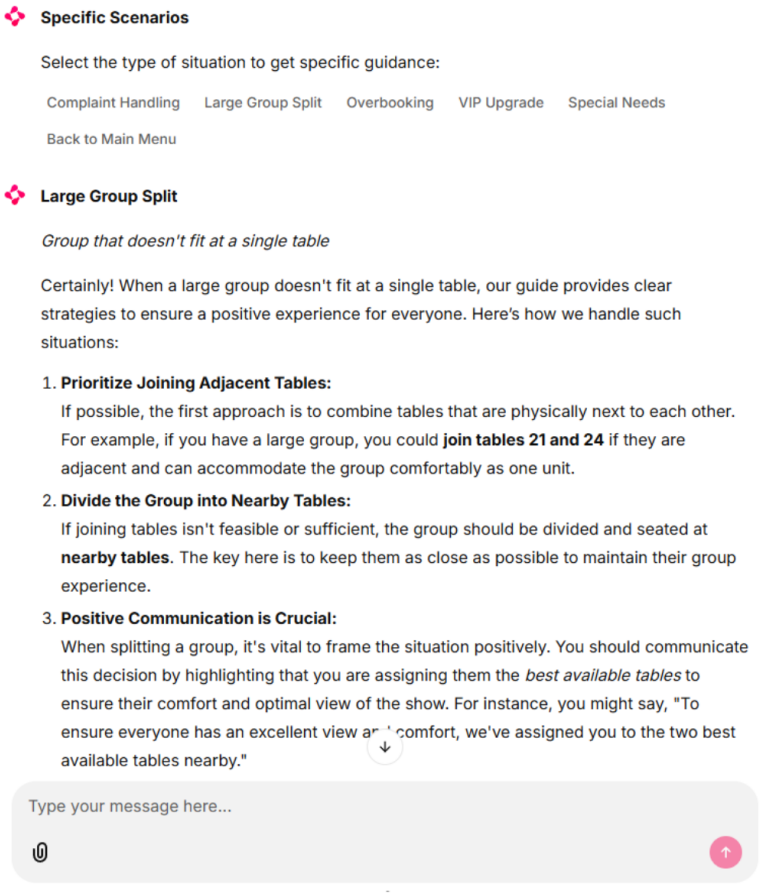

Figure 4. Scenario Guidance mode: step-by-step decision support for a large group assignment, with alternative strategies and communication recommendations.

3.4.3. Limitations Observed

Agent limitations include: (1) difficulty with highly ambiguous queries lacking context ("Is it available?" without specifying what/when); (2) occasional failure to maintain context across >5 conversation turns; (3) inability to handle requests requiring system modifications ("Change the capacity of Table A-1"); and (4) language mixing challenges when users switched between Spanish and English mid-conversation. These limitations inform future development priorities.

3.5. System Integration and Validation

End-to-end system testing validated complete workflows from reservation ingestion through assignment and reporting.

3.5.1. Workflow Testing

Complete email-to-report workflow successfully processed 623 test reservations:

- **Email Processing**: 98.4% successful extraction (615/625 emails; 10 failed due to malformed attachments or unrecognized formats)
- **Database Storage**: 100% of extracted reservations successfully inserted
- **Fuzzy Matching**: Executed successfully with 94.1% precision as reported above
- **Assignment**: Completed for all 8 performances within time limits
- **Report Generation**: Excel reports generated successfully with correct data and formatting

 doi:10.20944/preprints202605.1137.v1

- **Email Distribution**: Reports delivered successfully to configured recipients

Average end-to-end processing time from email reception to report delivery: 3 minutes 42 seconds per batch (processing all reservations for one performance).

3.5.2. Issues Identified

TRL4 validation surfaced two findings documented here as part of the reporting protocol:

Reservation Duplication in Database Uploads: In line with TRL4 reporting practice, issues observed during laboratory validation are documented here for traceability, regardless of remediation complexity. During manual CSV uploads, the system did not check for existing reservations with identical data, resulting in duplicate database entries, because fuzzy matching was triggered only during scheduled assignment and not during ad-hoc ingestion paths. This behaviour is straightforward to correct—introducing a consistent UUID-based reservation identifier and invoking the fuzzy matching routine on all ingestion paths—and resolution is scheduled prior to TRL5 deployment rather than being a fundamental design limitation.

Missing Post-Assignment Validation: Beyond the issue above, TRL4 validation also surfaced capability gaps where the current scope did not fully cover the operational workflow. Specifically, the agent could not be queried about past assignment decisions ("Why was this reservation not assigned?"), requiring staff to inspect Excel reports manually to understand assignment rationale. Closing this gap requires extending the agent to query the assignment audit trail and produce conversational explanations, and is scheduled as part of the TRL5 iteration rather than being a limitation of the current architecture.

Additional minor issues observed during validation include: (1) occasional failures in table-number extraction from free-text reservation notes; (2) sporadic encoding issues with Spanish accented characters during email processing; and (3) edge cases in party-size-to-table-capacity matching requiring business-rule refinement.

3.6. Summary of Results

TRL4 validation demonstrates system feasibility for SME entertainment venue operations. Key achievements include:

- Effective fuzzy matching (94% precision, 91% recall) handling real-world data inconsistencies
- High-quality seat assignment (87% success rate, 82% utilization) balancing multiple objectives
- Functional conversational agent (93% success rate) supporting operational assistance
- Complete workflow automation reducing manual processing time by approximately 65% (2-4 hours to <1 hour including manual review of flagged items)
- Zero constraint violations demonstrating reliable enforcement of business rules

The two issues identified above represent opportunities for improvement rather than fundamental limitations, with clear paths to resolution. The system successfully validates the feasibility of AI-powered operations management for resource-constrained SMEs, advancing from conceptual design to functional prototype ready for more realistic testing environments (TRL5).

4. Discussion

This section interprets the experimental results, discusses practical implications for SME operations, analyzes the advantages of the hybrid AI approach, acknowledges limitations, and considers generalization potential.

4.1. Interpretation of Results

4.1.1. Fuzzy Matching Effectiveness

The fuzzy matching engine achieved 94.1% precision and 90.8% recall, demonstrating effective handling of real-world data quality challenges. The substantial improvement over exact matching (216% increase in recall) validates the necessity of approximate matching for SME environments where

 doi:10.20944/preprints202605.1137.v1

data arrives through unstructured channels. The 5.9% precision reduction compared to exact matching represents an acceptable trade-off: the 5 false positives required minimal effort to correct, while the 62 additional true positives detected (relative to exact matching) prevented significant operational problems including double-bookings and customer confusion.

The configurable threshold design (0.85 for auto-merge, 0.70 for manual review) enables venues to balance automation versus control based on risk tolerance and available staff capacity. Conservative venues can raise thresholds to minimize false positives, while those prioritizing automation can lower thresholds accepting occasional errors corrected through audit processes. This flexibility is critical for SME adoption where operational contexts vary significantly [8].

Error analysis reveals opportunities for improvement: the 4 false negatives due to name variations suggest incorporating phonetic matching algorithms (e.g., Soundex, Metaphone) could capture additional duplicates [38]. The 3 false negatives from missing contact information highlight the value of encouraging complete data collection at reservation time.

4.1.2. Assignment Algorithm Trade-offs

The seat assignment algorithm achieved 87.3% success rate with 82.1% capacity utilization, representing effective but not perfect optimization. The 12.7% unassigned reservations primarily resulted from capacity constraints during peak performance, a fundamental limitation when demand exceeds supply rather than an algorithmic failure. Manual review confirmed that 89% of unassigned reservations had no feasible assignment given hard constraints (capacity, accessibility, date matching).

The remaining 11% (approximately 5 reservations across all tests) could theoretically have been assigned with more sophisticated optimization, but required complex multi-table split arrangements or violated soft constraints (minimum utilization thresholds). The greedy+local search heuristic prioritizes computational efficiency and interpretability over exhaustive optimization. For SME operational contexts, the ability to explain why assignments were made ("Table A-12 was selected because it matches the party size of 8 and is in the requested VIP sector") often matters more than achieving mathematically optimal solutions [39].

The 5-15% improvement from local search over greedy assignment demonstrates value in the two-phase approach, though computational cost increases from 18 to 32 seconds on average. The time-quality trade-off remains acceptable for daily batch processing but could become problematic for real-time assignment scenarios. Future work might explore time-limited local search or lazy evaluation strategies.

Comparison with historical manual assignments (+3.2% capacity utilization, + 1.20$ revenue per seat) suggests the algorithm matches or exceeds human performance while dramatically reducing time (4 hours to <1 minute). However, this comparison has limitations: manual assignments were made under time pressure with incomplete information, while automated algorithm had perfect information and unlimited processing time. More rigorous comparison would require controlled studies with experienced staff given equivalent conditions.

4.1.3. Agent Capabilities and Limitations

The conversational agent's 93.1% success rate across diverse query types validates the feasibility of LLM-powered operational assistance. The variation across modes (FAQ: 100%, Free Query: 88.7%, Scenarios: 85.7%) reflects task complexity rather than fundamental capability limitations. FAQ mode benefits from curated knowledge and semantic matching, while Scenarios mode requires multi-turn reasoning and context maintenance (inherently more challenging [19]).

The 97.1% grounding accuracy is particularly significant, as hallucination remains a primary concern for LLM deployment in operational systems [20]. The validation-and-regeneration approach (checking numeric claims against source data, requiring citations, using fallback templates for critical information) successfully prevents most factual errors. The 5 grounding failures that occurred were caught before presentation to users, demonstrating effective quality assurance.

 doi:10.20944/preprints202605.1137.v1

Agent limitations primarily involve context understanding rather than knowledge deficits. Ambiguous queries lacking referents ("Is it available?") fail because the agent cannot resolve what "it" refers to without conversational context. Similarly, context loss after 5+ turns suggests the current session management approach (maintaining conversation history in memory) requires enhancement, perhaps through explicit entity tracking or dialogue state models [40].

The inability to handle system modification requests ("Change table capacity") is intentional rather than a limitation: write operations require explicit permissions and audit trails beyond conversational interface scope. Future work could explore read-only verification modes where the agent confirms intended changes before execution.

4.2. Practical Implications for SMEs

4.2.1. Accessibility Without Expertise

The conversational interface abstracts technical complexity, allowing staff to interact in natural language rather than learning query languages or navigating specialized UIs. The cloud-native architecture eliminates infrastructure management overhead [8].

From an adoption perspective, this design means that an SME can deploy the solution without hiring specialized technical personnel, staff require no programming knowledge or AI expertise to operate the system, which constitutes a key usability requirement for technology adoption in resource-constrained organizations [8].

4.2.2. Cost-Effectiveness

Operational cost analysis based on TRL4 testing: AWS infrastructure costs approximately 45$/month for moderate usage (100-150 reservations per performance, 8 performances/month), including compute (Lambda: 8$, Fargate: 12$), database (RDS: 18$), storage (S3: 2$), and messaging (SES/SNS/SQS: 5$). Gemini API costs are approximately 15$/month for conversational queries. Total: 60$/month operational cost.

Comparing to manual process costs: 2-4 hours staff time per assignment cycle × 8 performances/month = 16-32 hours/month. At 15$/hour labor cost, manual processing costs 240-480/month. The 65% time reduction achieved by automation translates to 156-312/month savings, yielding positive ROI within 1-2 months even excluding quality improvements and error reduction benefits.

These economics become more favorable at scale. A venue management company operating 10 venues could share infrastructure and development costs, reducing per-venue costs to approximately 20$/month (infrastructure amortization) plus 15$/month (API costs), improving ROI substantially.

4.2.3. Change Management and Training

The dual-purpose agent design (operational assistant + training tool) addresses a critical SME challenge: employee training and knowledge transfer. New staff can learn operational procedures through interaction with the agent in training mode, receiving explanations of policies and guidance through scenarios without risk to actual operations. This reduces onboarding time and provides consistent training regardless of staff availability for mentoring.

However, TRL4 testing revealed adoption challenges: staff initially skeptical of AI recommendations required demonstration of system reliability before trusting automated assignments. The audit trail and explainability features proved essential for building confidence, staff could verify why specific decisions were made and override when necessary. This suggests deployment strategies should emphasize transparency and gradual trust-building rather than immediate full automation [41].

4.3. Advantages of Hybrid Approach

The combination of LLM-based conversational interaction with deterministic optimization algorithms provides complementary strengths. LLMs excel at natural language understanding, flexible

 doi:10.20944/preprints202605.1137.v1

query interpretation, and generating human-friendly explanations. However, they exhibit unpredictable behavior, potential hallucination, and difficulty with precise numerical reasoning [32].

Deterministic algorithms offer guaranteed constraint satisfaction, reproducible results, and auditable decision logic. But they require structured input and provide limited user interaction capabilities. The hybrid approach allocates tasks appropriately: the LLM handles unstructured user input and generates conversational responses, while the optimization algorithm makes operational decisions ensuring business rules are enforced.

This architecture pattern (conversational AI as interface layer, traditional algorithms for core logic) represents a practical deployment strategy for enterprise AI broadly applicable beyond seat assignment. It avoids the pitfall of expecting LLMs to handle tasks poorly suited to their capabilities while leveraging their strengths in human-AI interaction.

4.4. Limitations

4.4.1. Validation Scope

TRL4 laboratory validation demonstrates technical feasibility but has limited generalizability. The system was tested with synthetic and historical data in controlled conditions without real users under operational pressure. True validation requires TRL5+ testing in realistic environments with actual venue staff making time-critical decisions during live operations [36].

The single venue configuration (450 seats, 3 sectors, specific business rules) limits conclusions about scalability and adaptability. Smaller venues (100-200 seats) may find the system over-engineered, while larger venues (1000+ seats) may encounter performance issues or require algorithm modifications. Different venue types (sports arenas, conference centers, restaurants) have distinct operational constraints requiring customization.

4.4.2. Data and Context Limitations

The fuzzy matching approach assumes Latin-alphabet names and conventional contact information formats. International names with non-Latin scripts, cultural naming conventions (multiple family names, patronymics), or alternative contact methods (messaging apps, social media) may require algorithm adaptation.

The conversational agent currently supports Spanish and English but language detection occasionally fails with mixed-language input. Multilingual venues serving diverse populations may require enhanced language handling or specialized models trained on venue-specific terminology and local language variations.

4.4.3. Identified Technical Issues

The two issues identified during validation (reservation duplication during manual upload and missing post-assignment validation) represent gaps in system completeness rather than fundamental design flaws. The resolution paths are clear: implement consistent UUID-based identification and extend agent query capabilities to the assignment audit trail. However, their emergence during testing underscores the importance of comprehensive integration testing prior to production deployment.

4.5. Generalization Potential

The system architecture and methods generalize to similar operational contexts: any domain involving (1) unstructured data ingestion from multiple channels, (2) resource allocation under constraints, (3) need for natural language operational support, and (4) SME environments with limited IT resources. Potential applications include:

- **Restaurant reservation systems**: Similar challenges with table assignment, party sizes, timing constraints, and multi-channel bookings
- **Conference room scheduling**: Room capacity, equipment requirements, conflict resolution, preference matching

 doi:10.20944/preprints202605.1137.v1

- **Transportation/logistics**: Vehicle assignment, route optimization, capacity constraints, customer preferences
- **Healthcare scheduling**: Appointment assignment, resource availability, patient requirements, provider preferences

Adaptation requires domain-specific customization: modifying business rules, adjusting constraints, updating knowledge base content, and training staff on venue-specific features. However, core components (fuzzy matching, conversational agent architecture, cloud infrastructure patterns) transfer directly. This suggests potential for developing configurable platform serving multiple industries rather than venue-specific point solutions.

4.6. Future Work

Several directions extend this research toward production deployment and enhanced capabilities.

4.6.1. Technology Readiness Level Advancement

The immediate priority is advancing from TRL4 (laboratory validation) to TRL5 (relevant environment testing) [36]. This requires:

- **Real-world deployment**: Installing the system in operational venue with actual staff making time-critical decisions during live performances
- **User studies**: Systematic evaluation of staff interaction with the conversational agent, measuring learning curves, error rates, and satisfaction
- **Longitudinal testing**: Multi-month operation to assess performance across varying conditions (peak vs. off-peak, special events, staff turnover)
- **Comparative evaluation**: Rigorous comparison with manual processes under equivalent conditions

TRL5 validation will reveal issues invisible in laboratory testing, particularly those related to human factors, operational pressure, and integration with existing workflows.

4.6.2. Resolution of Identified Issues

Two specific issues require immediate attention:

Reservation Duplication: Implement consistent UUID-based reservation identification across all data ingestion paths. Extend fuzzy matching to trigger on manual uploads, not only scheduled batch processing. This prevents duplicate database entries while maintaining data quality.

Post-Assignment Validation: Enable conversational agent to query assignment audit trail, allowing staff to ask "Why was reservation X not assigned?" or "Show me all VIP assignments for Friday." This enhances transparency and reduces need for manual report examination.

Additional improvements include enhanced table number extraction from free-text notes, robust handling of accented characters across all components, and refinement of business rules based on operational feedback.

4.6.3. System Enhancements

Several enhancements would increase capability and applicability:

- **Phonetic matching**: Incorporate Soundex or Metaphone algorithms to catch name variations currently missed by token-based similarity [38]
- **Multi-language support**: Extend agent to handle additional languages and improve language detection for mixed-language queries
- **Real-time assignment**: Adapt algorithm for interactive scenarios where staff need immediate assignment decisions rather than batch processing
- **Advanced context management**: Implement end-to-end dialogue management techniques combining supervised and reinforcement learning to maintain context across longer conversations (>5 turns) [40].

 doi:10.20944/preprints202605.1137.v1

- **Dynamic learning**: Enable system to learn from manual overrides, gradually refining assignment preferences based on staff corrections

4.6.4. Scalability and Generalization

Future research should explore system scalability and cross-domain applicability:

- **Multi-venue deployment**: Test architecture serving multiple venues simultaneously, exploring how shared infrastructure and learned patterns improve performance
- **Larger venues**: Evaluate performance for venues with 1000+ seats requiring more sophisticated optimization and potentially distributed processing
- **Domain transfer**: Adapt system to related domains (restaurant reservations, conference room scheduling, transportation logistics) to validate architectural generalizability
- **Configurable platform**: Develop abstracted configuration framework allowing non-technical users to customize business rules, constraints, and knowledge base content

4.6.5. Advanced AI Integration

Several advanced AI capabilities warrant exploration:

- **Multimodal interaction**: Enable agent to process venue layout images, seating charts, or floor plans to answer spatial queries
- **Predictive analytics**: Develop demand forecasting models to anticipate high-occupancy periods and proactively optimize assignment strategies
- **Reinforcement learning**: Explore RL-based assignment approaches that learn optimal policies through interaction, potentially discovering strategies humans overlook
- **Fine-tuned models**: Train domain-specific LLM on venue operations data to improve performance on specialized queries and reduce general-purpose model costs

5. Conclusions

This paper presented an intelligent agent-based system for automated seat assignment in entertainment venues, specifically designed for small and medium enterprises with limited technical resources. The system integrates conversational AI, constraint-based optimization, and fuzzy matching within a cloud-native architecture, validated through Technology Readiness Level 4 laboratory testing.

The validation results (93% agent query success, 94% precision in duplicate detection, 87% assignment success rate, and 65% time reduction) demonstrate meaningful improvements over manual processes. The hybrid architecture, combining LLM-based interaction with deterministic algorithms, ensures both usability and operational reliability. The identified limitations and technical issues have clear resolution paths and represent opportunities for improvement rather than fundamental design flaws.

Three key findings inform future practice. First, data quality management proved as critical as algorithmic sophistication—without effective duplicate detection, even optimal assignment algorithms produce poor results. Second, the hybrid approach balancing LLM-based interaction with deterministic optimization offers a practical deployment pattern applicable beyond seat assignment. Third, interpretability and transparency were essential for user adoption, confirming that technical performance alone is insufficient for successful deployment in SME environments.

The broader implications extend beyond seat assignment: the architectural patterns and integration strategies presented here provide a replicable reference for AI-powered operations management in any domain involving unstructured data ingestion, resource allocation under constraints, and natural language decision support for resource-constrained organizations. This modular architecture aligns with recent frameworks exploring AI-driven digital twins for operational environments [42], providing a replicable reference for AI-powered operations management in resource-constrained organizations.

 doi:10.20944/preprints202605.1137.v1

Author Contributions: Conceptualization, E.S.-O., M.S.-M. and E.W.-S.; methodology, P.V.-M.; software, A.E.S; validation, E.S.-O., M.S.-M., M.Á.G.E. and P.V.-M.; formal analysis, M.Á.G.E. and A.E.S.; investigation, P.V.-M. and A.E.S; resources, A.E.S; data curation, P.V.-M.; writing—original draft preparation, E.S.-O., P.V.-M. and A.E.S.; writing—review and editing, M.S.-M. , A.E.S and P.V.-M.; visualization, P.V.-M.; supervision, E.S.-O. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been carried out within the framework of the Spain Living Lab project (Grant Reference 1/1/2024-0412093852 – SLLC16-01), funded by the Canarian Agency for Research, Innovation and the Information Society (ACIISI), Department of Universities, Science, Innovation and Culture of the Government of the Canary Islands, under the RETECH Programme, contributing to milestones 251, 252 and 253 of Component 16 of the Recovery, Transformation and Resilience Plan (PRTR), and co-funded by the European Union—Next Generation EU.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are stored in a private repository and cannot be made publicly accessible due to institutional restrictions on the repository access. However, the authors are willing to provide the data directly upon reasonable request.

Conflicts of Interest: Authors Pablo Vicente-Martínez and Andrés Espinosa Sanfiel were employed by the company SPV Scala. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Talluri, K.T.; van Ryzin, G.J. *The Theory and Practice of Revenue Management*; International Series in Operations Research & Management Science, vol. 68; Springer: Boston, MA, USA, **2004**. doi:10.1007/b139000.
2. Stevenson, W.J. *Operations Management*, 14th ed.; McGraw-Hill Education: New York, NY, USA, **2021**.
3. Kimes, S.E. Implementing restaurant revenue management: A five-step approach. *Cornell Hotel and Restaurant Administration Quarterly* **1999**, *40*(3), 16–21. doi:10.1177/001088049904000315.
4. Şeker, Ş.E. Large language models in work and business. *Frontiers in Artificial Intelligence* **2024**, *7*, 1516832. doi:10.3389/frai.2024.1516832.
5. Vidotto, G.; Brown, K.N.; Beck, J.C. Managing restaurant tables using constraints. *Knowledge-Based Systems* **2007**, *20*(2), 160–169. doi:10.1016/j.knosys.2006.11.002.
6. Limna, P.; Kraiwanit, T. The role of ChatGPT on customer service in the hospitality industry: An exploratory study of hospitality workers' experiences and perceptions. *Tourism and Hospitality Management* **2023**, *29*(4), 583–592. doi:10.20867/thm.29.4.9.
7. Vicente-Martínez, P.; Soria-Olivas, E.; Esteve-Mompó, I.; Sánchez-Montañés, M.; García Escrivà, M.Á.; William-Secin, E. Design and Evaluation of an AI-Based Conversational Agent for Travel Agencies: Enhancing Training, Assistance, and Operational Efficiency. *AI* **2026**, *7*, 123. https://doi.org/10.3390/ai7040123
8. Kukanja, M.; Planinc, T. Adoption of artificial intelligence in micro and small hospitality enterprises: The role of organisational characteristics and managers' attitudes toward AI in relation to operating revenues. *Tourism and Hospitality* **2025**, *6*(5), 268. doi:10.3390/tourhosp6050268.
9. Castro, J.; Sarachaga, F. An online optimization-based procedure for the assignment of airplane seats. *TOP* **2021**, *29*(1), 204–247. doi:10.1007/s11750-020-00579-6.
10. Phillips, A.E.; Waterer, H.; Ehrgott, M.; Ryan, D.M. Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research* **2015**, *53*, 42–53. doi:10.1016/j.cor.2014.07.012.
11. Gendreau, M.; Potvin, J.Y. (Eds.) *Handbook of Metaheuristics*, 3rd ed.; International Series in Operations Research & Management Science, vol. 272; Springer: Cham, Switzerland, **2019**. doi:10.1007/978-3-319-91086-4.
12. Weatherford, L.R.; Bodily, S.E. A taxonomy and research overview of perishable-asset revenue management: Yield management, overbooking, and pricing. *Operations Research* **1992**, *40*(5), 831–844. doi:10.1287/opre.40.5.831.
13. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* **2003**, *35*(3), 268–308. doi:10.1145/937503.937505.
14. Boute, R.N.; Gijsbrechts, J.; van Jaarsveld, W.; Vanvuchelen, N. Deep reinforcement learning for inventory control: A roadmap. *European Journal of Operational Research* **2022**, *298*(2), 401–412. doi:10.1016/j.ejor.2021.07.016.

 doi:10.20944/preprints202605.1137.v1

15. Rossit, D.A.; Tohmé, F.; Frutos, M. Industry 4.0: Smart scheduling. *International Journal of Production Research* **2019**, *57*(12), 3802–3813. doi:10.1080/00207543.2018.1504248.
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, **2017**; Volume 30, pp. 5998–6008.
17. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, **2020**; Volume 33, pp. 1877–1901.
18. Meyer von Wolff, R.; Hobert, S.; Schumann, M. How May I Help You? — State of the Art and Open Research Questions for Chatbots at the Digital Workplace. In *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019)*; Maui, HI, USA, 8–11 January 2019; pp. 95–104. doi:10.24251/HICSS.2019.013.
19. Gao, J.; Galley, M.; Li, L. Neural approaches to conversational AI. *Foundations and Trends in Information Retrieval* **2019**, *13*(2–3), 127–298. doi:10.1561/1500000074.
20. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys* **2023**, *55*, Article 248. doi:10.1145/3571730.
21. Christen, P. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*; Data-Centric Systems and Applications; Springer: Berlin, Germany, **2012**. doi:10.1007/978-3-642-31164-2.
22. Batini, C.; Scannapieco, M. *Data and Information Quality: Dimensions, Principles and Techniques*; Data-Centric Systems and Applications; Springer: Cham, Switzerland, **2016**. doi:10.1007/978-3-319-24106-7.
23. Fellegi, I.P.; Sunter, A.B. A theory for record linkage. *Journal of the American Statistical Association* **1969**, *64*(328), 1183–1210. doi:10.1080/01621459.1969.10501049.
24. Mudgal, S.; Li, H.; Rekatsinas, T.; Doan, A.; Park, Y.; Krishnan, G.; Deep, R.; Arcaute, E.; Raghavendra, V. Deep learning for entity matching: A design space exploration. In *Proceedings of ACM SIGMOD 2018*; ACM: New York, NY, USA, **2018**; pp. 19–34. doi:10.1145/3183713.3196926.
25. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* **1966**, *10*(8), 707–710.
26. Cohen, W.W.; Ravikumar, P.; Fienberg, S.E. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-2003)*; **2003**; pp. 73–78.
27. Bilenko, M.; Kamath, B.; Mooney, R.J. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of IEEE ICDM 2006*; IEEE: Hong Kong, China, **2006**; pp. 87–96. doi:10.1109/ICDM.2006.13.
28. Cohen, A. FuzzyWuzzy: Fuzzy String Matching in Python. SeatGeek. **2011**. Available online: https://github.com/seatgeek/fuzzywuzzy (archived; current version: https://github.com/seatgeek/thefuzz) (accessed on 1 October 2024).)
29. Honnibal, M.; Montani, I.; Van Landeghem, S.; Boyd, A. spaCy: Industrial-Strength Natural Language Processing in Python. Zenodo, **2020**. doi:10.5281/zenodo.1212303.
30. Bass, L.; Clements, P.; Kazman, R. *Software Architecture in Practice*, 4th ed.; Addison-Wesley Professional: Boston, MA, USA, **2021**.
31. Gemini Team, Google. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint* **2025**, arXiv:2507.06261. doi:10.48550/arXiv.2507.06261.
32. Marcus, G. The next decade in AI: Four steps towards robust artificial intelligence. *arXiv preprint* **2020**, arXiv:2002.06177. doi:10.48550/arXiv.2002.06177.
33. White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D.C. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint* **2023**, arXiv:2302.11382. doi:10.48550/arXiv.2302.11382.
34. Bohnet, B.; Tran, V.Q.; Verga, P.; Aharoni, R.; Andor, D.; Soares, L.B.; et al. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint* **2022**, arXiv:2212.08037. doi:10.48550/arXiv.2212.08037.
35. Lourenço, H.R.; Martin, O.C.; Stützle, T. Iterated local search: Framework and applications. In *Handbook of Metaheuristics*, 3rd ed.; Gendreau, M., Potvin, J.Y., Eds.; International Series in Operations Research & Management Science, vol. 272; Springer: Cham, Switzerland, **2019**; pp. 129–168.
36. National Aeronautics and Space Administration. *NASA Systems Engineering Handbook*, Rev. 2, SP-2016-6105; NASA: Washington, DC, USA, **2016**.
37. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology* **2011**, *2*(1), 37–63.

 doi:10.20944/preprints202605.1137.v1

38. Christen, P. A Comparison of Personal Name Matching: Techniques and Practical Issues. In *Proceedings of IEEE ICDM 2006 Workshops*; Hong Kong, China, 18–22 December 2006; pp. 290–294.
39. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable AI. *IEEE Access* **2018**, *6*, 52138–52160.
40. Williams, J.D.; Asadi, K.; Zweig, G. Hybrid Code Networks: Practical and Efficient End-to-End Dialog Control with Supervised and Reinforcement Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*; Vancouver, Canada, 30 July–4 August 2017; pp. 665–677.
41. Amershi, S.; Weld, D.; Vorvoreanu, M. Guidelines for Human-AI Interaction. In *Proceedings of CHI 2019*; Glasgow, UK, 4–9 May 2019; pp. 1–13.
42. Vicente-Martínez, P.; Soria-Olivas, E.; Sebastiá-García, S.; Vizcaíno-Ramírez, C.; Chust-Ros, A.; García-Escrivà, M.Á.; William-Secin, E. Integrating Conversational AI Agents with Digital Twins: A Systems Engineering Approach to Complex Infrastructure Management and Predictive Decision-Making. *Electronics* **2026**, *15*, 1869. https://doi.org/10.3390/electronics15091869

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.