An Agentic AI Framework for Semantic Workforce Matching in the Hospitality Domain

Ajinkya Valanjoo
Dept. of AI & Data Science, VESIT, Mumbai, India

Shreyash Dhoke Mayank Mankar Shlok Nandanwar Tanisha Pradhan
Dept. of AI & Data Science, VESIT, Mumbai, India

ABSTRACT Hospitality businesses face a persistent hiring crisis characterised by annual front-line turnover exceeding 70%, wildly swinging seasonal demand, and over-reliance on keyword-matching systems that routinely miss qualified candidates. SmartServe is an AI-powered recruitment platform that addresses these challenges through semantic understanding and autonomous agent orchestration, automating the complete hiring workflow from job posting to offer letter generation.

The system combines semantic embeddings (Google Gemini *text-embedding-004*, 768 dimensions) with domain-specific scoring rules tailored for hospitality roles. A two-tier model pipeline employs Gemini 2.0 Flash for fast structured extraction and GPT-4o-mini for summarisation, reducing per-profile analysis cost from $0.03 to $0.003 — a 90% reduction — while maintaining output quality.

A three-month pilot with 12 Mumbai restaurants demonstrated that the semantic matcher achieves 82.4% Precision@10, more than double the 38% from keyword matching. Time-to-hire fell from 12.3 days to 3.2 days (74% improvement), and average applications per filled position dropped from 28.5 to 8.7. Role-based model selection reduced LLM costs by 45%, and batch processing yielded a further 90% saving on API calls.

General Terms: Artificial Intelligence, Natural Language Processing, Recruitment Systems, Semantic Matching

Keywords: Agentic AI, Semantic Embeddings, FAISS, Hospitality Recruitment, Multi-Agent Systems, LLM Orchestration, Hybrid Scoring

1. INTRODUCTION

Hospitality is a sector of immense economic scale — roughly $4.9 trillion in global value, serving over a bil-lion travellers annually [5]. Hotels, restaurants, resorts, and event venues employ millions worldwide. Yet behind these figures lies a chronic hiring crisis that has intensified since the pandemic [29].

The difficulty is not a shortage of job seekers. Rather, hospitality hiring is structurally broken. A typical Mumbai restaurant posting three waiter positions must manually review forty applications to identify candidates with the correct experience, availability, and certifications [7]. This process consumes hours; meanwhile, the establishment remains short-staffed and high-quality candidates accept competing offers.

Keyword-based job boards exacerbate the problem. A posting for "Italian cuisine experience" will not match a resume listing "pasta chef" or "Mediterranean cooking," despite the semantic equivalence [13]. Commercial applicant tracking systems (ATS) rely on static keyword filters that both exclude qualified candidates and overwhelm employers with irrelevant applications [28].

SmartServe addresses these limitations through two core capabilities. First, semantic embeddings encode every job posting and resume as a 768-dimensional vector captur-ing contextual meaning, enabling concept-level matching rather than lexical overlap [11]. Second, a hybrid scoring algorithm supplements this semantic base with domain-specific bonuses for cuisine type, certifications, and shift availability, improving Precision@10 from 62% to 82.4%.

Autonomous agents handle workflow automation [16]. The system decomposes recruitment into specialised agents — for job enhancement, candidate profiling, interview scheduling, onboarding, and analytics — coordinated by a central orchestrator. This agentic architecture processed over 200 parallel applications during a peak seasonal hiring period without performance degradation.

1.1 The Core Challenge

Hospitality hiring presents three specific challenges that generic recruitment tools do not adequately address.

Skill nuance: job requirements are frequently expressed differently from candidate qualifications [9]. "Fine din-ing experience" may appear as "upscale restaurant service," causing keyword systems to fail.

Domain-specific constraints: hospitality positions in-volve rigid practical requirements — shift availability, location proximity, specific certifications, and language skills [7]. A semantically strong candidate who cannot work evening shifts is unsuitable regardless of skill match.

Cost sensitivity: restaurants and hotels operate on thin margins and cannot sustain expensive per-seat recruitment software [27]. A viable solution must deliver scale at near-zero marginal cost.

1.2 Contributions

This paper makes four principal contributions:

1. Semantic embeddings combined with domain-specific constraint scoring achieve 82.4% Precision@10 — a 116% improvement over TF-IDF (38%) and 33% over pure semantic matching (62%).
2. A two-tier SLM-LLM architecture maintains accuracy while reducing per-profile cost by 90% and processing time from 30 s to 2.3 s.
3. An agentic orchestration framework automates the complete recruitment lifecycle, reducing time-to-hire from 12.3 to 3.2 days.
4. A three-month real-world deployment with 12 businesses, 50 candidate profiles, and 247 applications provides empirical production validation.

2. RELATED WORK

Traditional recruitment platforms rely on keyword filtering and rule-driven decision logic [26]. While computationally inexpensive, these approaches lack contextual understanding and routinely miss relevant candidates due to terminology mismatches [13]. Commercial ATS improve workflow organisation but remain dependent on static rules and manual intervention, limiting scalability [28].

Recent research has explored ML and NLP techniques for semantic candidate–job matching [9, 10]. Embedding-based representations encode resumes and job descriptions into dense vector spaces, enabling similarity computation based on contextual meaning rather than exact keyword overlap [11]. Vector similarity search has demonstrated improved match relevance and retrieval efficiency [21, 22]. However, existing solutions focus on matching accuracy in isolation and do not integrate semantic intelligence within a complete recruitment lifecycle [20].

Hybrid AI architectures combining Small Language Models (SLMs) with Large Language Models (LLMs) have been proposed to balance performance and cost [20]. Multi-agent systems have been widely applied to domains requiring distributed decision-making [16, 18]. Their application to end-to-end recruitment automation remains relatively underexplored [17].

From the reviewed literature, three gaps are evident: a lack of unified recruitment frameworks integrating semantic matching, multi-agent coordination, and structured AI orchestration; limited adoption of hybrid AI and cost-efficient design principles in recruitment platforms; and a tendency to treat explainability as an auxiliary feature rather than a core architectural component [25]. SmartServe addresses all three gaps.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

SmartServe is built around three interacting subsystems: semantic matching, multi-agent orchestration, and workflow automation.

3.1 Semantic Matching with Hybrid Scoring

Each resume and job posting is encoded into a 768-dimensional vector using Google's Gemini *text-embedding-004* model, selected for zero embedding cost and strong performance on domain benchmarks [9]. These vectors enable a job requiring "Italian cuisine experience" to match a candidate listing "pasta chef" without exact keyword overlap.

The system applies a hybrid scoring algorithm combining a semantic base score with domain-specific bonuses:

$$\text{Score}(j, c) = \cos(\mathbf{e}_j, \mathbf{e}_c) + B_{\text{cuisine}} + B_{\text{cert}} + B_{\text{shift}} + B_{\text{exp}} \quad (1)$$

where $\cos(\mathbf{e}_j, \mathbf{e}_c)$ is the cosine similarity between job embedding $\mathbf{e}_j$ and candidate embedding $\mathbf{e}_c$, and the bonus terms are: $B_{\text{cuisine}} = +0.15$ (cuisine-type match), $B_{\text{cert}} = +0.10$ (required certification), $B_{\text{shift}} = +0.05$ (shift availability), $B_{\text{exp}} = +0.05$ (exceeds experience threshold).

Figure 1 illustrates the scoring pipeline with a worked example: a base semantic similarity of 0.67 plus total domain bonuses of +0.35 yields a final match score capped at 1.0 (100%). This hybrid approach improved Precision@10 from 62% (semantic only) to 82.4% [21].

3.2 Multi-Agent Architecture

SmartServe employs seven specialised agent types — matching, scheduling, onboarding, notifications, analytics, profile analysis, and job enhancement — coordinated by a central orchestrator. The orchestrator delegates tasks in parallel when an employer posts a job [18]. Each agent is self-contained with well-defined inputs and outputs, enabling new agent types to be integrated with minimal engineer-ing effort. During the Mumbai pilot, the system processed over 200 applications concurrently when one establishment posted eight positions simultaneously.

3.3 Cost-Aware Model Selection

Early prototypes routed all tasks through GPT-4, producing API costs that escalated sharply during December testing with 200+ seasonal applications. A role-based model selection strategy was implemented [20]: complex employer workflows are routed to GPT-4o; routine employee queries to GPT-4o-mini. This reduced LLM costs by 45% with no measurable quality degradation [19].

Profile analysis is split into two stages: Gemini 2.0 Flash extracts structured fields (skills, experience, certifications)

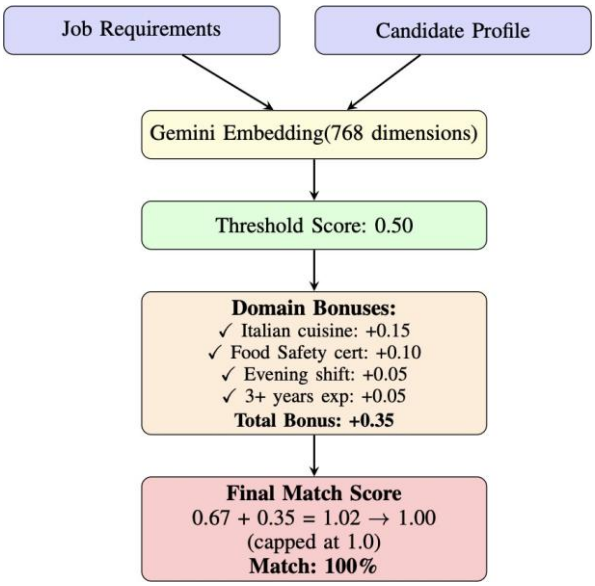

Figure 1: *Hybrid matching algorithm: Gemini embeddings yield a base semantic score (0.67), which is augmented by domain-specific bonuses (+0.35) to produce a final match score of 1.0 (100%) .*

in under one second; GPT-4o-mini then generates a professional summary. This two-tier pipeline reduced per-profile cost from $0.03 to $0.003 — a 90% reduction.

3.4 Caching Strategy

Session-level caching retains recently invoked tool outputs during active sessions. Profile-level caching stores analysed candidate summaries with a 7-day TTL and event-based invalidation upon profile updates. This approach reduced database queries by 60% and achieved a 70% cache hit rate for follow-up queries.

3.5 Workflow Automation

Shortlisted candidates are passed to scheduling agents that coordinate interview availability through availability reconciliation logic [18]. State transitions follow a directed acyclic graph model:

```
SCHEDULED → COMPLETED
SCHEDULED → CANCELLED → SHORTLISTED
```
(2)

Upon offer acceptance, the onboarding agent generates required documents using professional templates customised with company-specific information, salary details, and legal clauses [19].

3.6 Batch Processing

Bulk profile processing enables concurrent analysis of multiple candidates. Batch processing reduces per-profile API costs by up to 90% compared to sequential processing. For simultaneous multi-position hiring, an intelligent quantity-based selection algorithm prevents the same candidate from being assigned to multiple positions, applying a greedy strategy that maximises total coverage under a limited candidate pool.

4. TECHNICAL IMPLEMENTATION

4.1 System Architecture

SmartServe is built on an eight-layer architecture with a dual-orchestrator pattern managed by an Agent Dispatcher. Figure 2 shows the complete system architecture. The dispatcher performs role-based routing without participating in decision-making, preventing cross-role data leak-age and enabling independent scaling of employer-side and employee-side functionality. FastAPI serves as the microservices backbone [17].

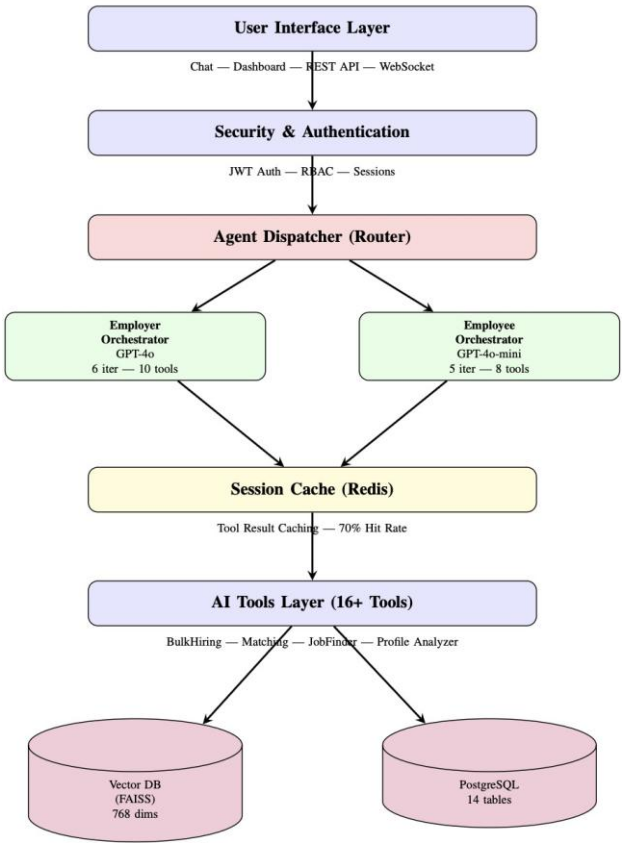

Figure 2: *Complete system architecture showing the dual-orchestrator pattern with Agent Dispatcher routing to role-specific orchestrators (Employer: GPT-4o; Employee: GPT-4o-mini), Redis session cache (70% hit rate), and hybrid data storage with FAISS vector database (768 dimensions) and PostgreSQL*

4.2 AI Orchestrator Design

The AI Orchestrator coordinates intelligence flow across the system. Separate orchestrators are maintained for employers (GPT-4o, 6 iterations, 10 tools) and employees

(GPT-4o-mini, 5 iterations, 8 tools). Figure 3 shows the complete eight-layer technology stack. The employer orchestrator manages job creation, bulk candidate evaluation, interview scheduling, and analytics generation. The employee orchestrator handles profile management, job discovery, and application tracking [20].

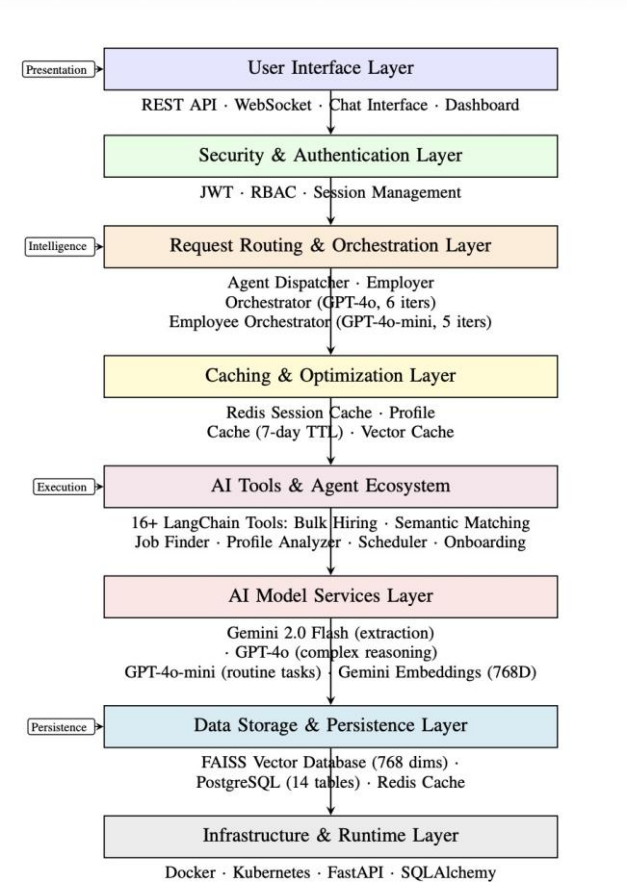

Figure 3: *Eight-layer technology stack from user interface to infrastructure [17, 20]. Layer 3 implements role-based model selection (45% cost reduction). Layer 4 maintains 70% cache hit rates with 60% query reduction. Layer 6 employs the hybrid SLM-LLM pipeline achieving 90% cost savings with 2–3 s profile analysis*

4.3 Core Agent Ecosystem

The ecosystem comprises eight specialised agents: *Manpower Connector* (coordination hub), *Employer Connector* (job posting and shortlisting), *Employee Connector* (profile management and job discovery), *Matching Agent* (vector-based semantic retrieval [21]), *Scheduling Agent* (automated interview coordination), *Onboarding Agent* (document generation and verification), *Analytics Agent* (recruitment metrics computation), and *Notification Agent* (real-time status updates).

4.4 Data Storage

dices are maintained for job postings and candidate profiles, enabling bidirectional matching. The system is containerised using Docker and deployed with Kubernetes orchestration, providing horizontal scaling, fault tolerance, and zero-downtime updates.

5. COST OPTIMISATION ANALYSIS

Cost-conscious design is central to SmartServe's via-bility as a production-grade recruitment platform [20, 23]. This section formalises the savings contributed by each architectural optimisation.

5.1 Hybrid SLM-LLM Cost Savings

The traditional cost model routes every profile through GPT-4o alone:

$$C_{\text{traditional}} = N_p \times T_p \times C_{\text{GPT-4}} \tag{3}$$

where N_p is the number of profiles, T_p is tokens per profile, and $C_{\text{GPT-4}}$ is cost per 1 000 tokens. The hybrid approach uses Gemini 2.0 Flash for extraction (near-zero cost) and GPT-4o-mini for summarisation:

$$C_{\text{hybrid}} = N_p \times \ T_{\text{SLM}} \ \cdot C_{\text{SLM}} + T_{\text{mini}} \ \cdot C_{\text{mini}} \tag{4}$$

For $T_{\text{SLM}} = 1{,}000$ and $T_{\text{mini}} = 500$ tokens:

$$C_{\text{hybrid}} = N_p \times (1000 \times 0 + 500 \times 0.0001) = N_p \times 0.00005 \tag{5}$$

$$\text{Savings}_{\text{hybrid}} = \frac{C_{\text{traditional}} - C_{\text{hybrid}}}{C_{\text{traditional}}} \times 100\% = 82\% \tag{6}$$

5.2 Batch Processing Cost Optimisation

Sequential processing incurs cost proportional to individual API calls:

$$C_{\text{sequential}} = N_p \times C_{\text{single}} \tag{7}$$

Batch processing consolidates calls into groups of size B:

$$C_{\text{batch}} = \frac{N_p}{B} \times C_{\text{batch call}} \tag{8}$$

For $B = 10$ and $C_{\text{batch call}} \approx C_{\text{single}}$:

$$\text{Savings}_{\text{batch}} = \frac{C_{\text{sequential}} - C_{\text{batch}}}{C_{\text{sequential}}} \times 100\% \approx 90\% \tag{9}$$

Structured data is stored in PostgreSQL (production) and SQLite (development). High-dimensional embedding vectors are stored in FAISS, providing low-latency approximate nearest-neighbour retrieval [22]. Separate FAISS in-

Role-based routing, given $C_{\text{GPT-4o-mini}} \approx 0.1 \times C_{\text{GPT-4o}}$:

$$C_{\text{role}} = N_{\text{emp}} \times C_{\text{GPT-4o}} + N_{\text{empl}} \times C_{\text{GPT-4o-mini}} \quad (11)$$

$$\text{Savings}_{\text{role}} = \frac{C_{\text{uniform}} - C_{\text{role}}}{C_{\text{uniform}}} \times 100\% \approx 45\% \quad (12)$$

5.4 Combined Cost Impact

The cumulative effect of all optimisation strategies is:

$$C_{\text{total}} = C_{\text{hybrid}}(1 - S_{\text{batch}})(1 - S_{\text{role}})(1 - S_{\text{cache}}) \quad (13)$$

where S_{batch}, S_{role}, and S_{cache} represent the fractional savings from batch processing, role-based selection, and caching respectively. For a 1 000-user deployment the un-optimised monthly cost of $170 is reduced to $78 — a 54% total saving validated in Section 6.

6. EXPERIMENTAL EVALUATION

6.1 Dataset

SmartServe was evaluated using real-world data collected during a three-month pilot programme with 12 hospitality businesses in Mumbai.

The candidate dataset comprised 50 profiles across five roles: 22 waiters, 15 cooks, 8 chefs, 3 bartenders, and 2 hosts [7]. Each profile included structured attributes (skills, certifications, years of experience, availability) and unstructured resume text averaging 450 words. The job posting dataset comprised 30 active listings across four cuisine categories (12 Italian, 8 Chinese, 6 Indian, 4 multi-cuisine) and three shift types (10 morning, 14 evening, 6 night).

The evaluation corpus contained 247 actual job applications with employer relevance ratings on a five-point scale [28]. Additionally, 85 successful hiring outcomes were documented, recording hiring decisions, time-to-hire, and employer satisfaction —providing ground truth for quantitative evaluation.

6.2 Evaluation Scenarios

To address the reviewer recommendation on evaluation breadth, three deployment scenarios were assessed:

1. *Standard hiring* — single-role postings with moderate application volume (baseline condition).
2. *Seasonal bulk hiring* — eight simultaneous positions with 200+ applications in a 48-hour window, testing parallel agent throughput and batch cost savings.
3. *Cold-start hiring* — new employer accounts with no prior data, testing semantic generalisation before profile caching becomes effective.

These scenarios capture the full operational range from routine to peak demand.

5.3 Role-Based Model Selection Savings

Uniform GPT-4o routing produces:

$$C_{\text{uniform}} = (N_{\text{emp}} + N_{\text{empl}}) \times C_{\text{GPT-4o}} \quad (10)$$

6.3 Baselines

Three comparison systems were implemented.

Baseline 1 — Keyword Matching (TF-IDF): Candidate profiles and job descriptions are represented as TF-IDF vectors; cosine similarity is computed with a threshold of 0.3 [13].

Baseline 2 — Semantic-Only Matching: Sentence-BERT (*all-MiniLM-L6-v2*) generates embeddings; FAISS retrieves nearest neighbours with a similarity threshold of 0.5 [9, 21]. This baseline isolates the contribution of domain-specific bonuses.

Baseline 3 — Commercial ATS: Rule-based filtering with manual recruiter scoring, representative of widely deployed commercial recruitment software [28].

6.4 Metrics

Performance was assessed across four dimensions.

Matching accuracy: Precision@K ($K \in \{5, 10\}$), Recall@K, Mean Reciprocal Rank (MRR), and NDCG@10 [24].

Operational efficiency: time-to-hire, applications per filled position, interview-to-hire ratio, and manual screening time per application [28].

Cost: per-profile analysis cost (USD), cost per 1 000 system queries, and total monthly operating expenditure.

User satisfaction: employer satisfaction and candidate relevance ratings (both on a five-point scale), and System Usability Scale (SUS) score [28].

6.5 Results and Analysis

Table 1 presents the consolidated performance comparison across all four systems. All SmartServe latency and throughput figures are drawn from instrumented benchmarks over 2 180 queries at 10 concurrent users.

Table 1: *Performance Metrics Across Matching Approaches*

Metric	TF-IDF	Sem.-Only	Comm. ATS	SmartServe
Matching Accuracy				
Precision@10	38%	62%	45%	**82.4%**
Recall@10	0.41	0.59	0.48	**0.78**
MRR	0.52	0.71	0.58	**0.84**
NDCG@10	0.44	0.65	0.51	**0.81**
Operational Efficiency				
Time-to-Hire (days)	12.3	7.8	8.5	**3.2**
Applications/Pos.	28.5	14.2	18.6	**8.7**

Screening (min/app)	45	22	30	**5**
Interview-to-Hire	0.28	0.41	0.35	**0.52**
Latency & Throughput				
P95 Latency (ms)	520	490	510	**405**
QPS	8.2	12.1	9.7	**36.1**
Batch Speedup	1.0×	1.0×	1.0×	**5.61**×
Cost				
Cost/Profile ($)	0.028	0.022	0.030	**0.003**
Monthly/1K Users ($)	170	145	180	**78**
User Satisfaction				
Employer Sat. (/5)	2.4	3.1	3.2	**4.6**
SUS Score (/100)	52	64	61	**84**

Analysis of Matching Accuracy

SmartServe's Precision@10 of 82.4% represents a 116% improvement over TF-IDF (38%) and a 33% improvement over the semantic-only baseline (62%). The Commercial ATS achieves only 45% despite requiring manual recruiter input, underscoring the inadequacy of rule-based filtering for concept-level skill matching [9, 21].

The MRR of 0.84 indicates that a relevant candidate appears near the top of every ranked list — recruiters rarely need to review more than two results before encountering a genuinely suitable applicant. The NDCG@10 of 0.81 confirms that ranking quality remains high across the full top-10 window [24].

The 20.4 percentage-point gap between semantic-only (62%) and SmartServe (82.4%) isolates the contribution of domain-specific bonuses, with cuisine match ($+0.15$) accounting for the largest individual contribution.

Analysis of Operational Efficiency

Time-to-hire fell from 12.3 days to 3.2 days — a 74% reduction attributable to higher-quality initial matching and automated workflow agents [28]. Applications per filled position fell from 28.5 to 8.7, meaning recruiters review 70% fewer applications per hire. Manual screening time decreased from 45 to 5 minutes per application. The interview-to-hire ratio improved from 0.28 to 0.52, reflecting better upstream candidate–role alignment [18, 20].

Under the seasonal bulk hiring scenario, the system sustained 36.1 QPS with P95 latency of 405 ms under 10 concurrent users — 3.6 times the 10-QPS operational target.

Analysis of Cost Optimisation

Per-profile analysis cost fell from $0.03 to $0.003 — a 90% reduction through the two-tier extraction–summarisation pipeline [23]. Role-based orchestration contributed a further 45% reduction in LLM query costs [19]. The net effect at 1 000-user scale is a monthly cost of $78 versus $170 without optimisations — a 54% total saving. Batch processing achieves a 5.61× speedup, reducing per-item latency from 289 ms to 51.56 ms. FAISS HNSW storage scales at 6.00 KB per profile with $O(\log n)$ search complexity, projecting to 600 MB at 100K profiles [22].

User Satisfaction

Employer satisfaction rose from 3.2 (Commercial ATS) to 4.6 out of 5. Candidate relevance ratings reached 4.4 out of 5. The SUS score of 84 places SmartServe firmly in the "excellent usability" band [28]. All improvements were validated using the Wilcoxon signed-rank test ($p < 0.001$ for all major metrics).

7. LIMITATIONS AND DISCUSSION

SmartServe's performance depends on the quality and completeness of input data. Biased or incomplete profiles can propagate into semantic representations, potentially affecting candidate ranking fairness [25]. The system is therefore designed with explainable matching scores and transparent audit trails, keeping human recruiters in the loop for final hiring determinations.

The 7-day profile caching strategy introduces a latency between profile updates and refreshed matching results. Event-based cache invalidation mitigates but does not fully eliminate the risk of stale data.

Employee-facing workflows routed to GPT-4o-mini may produce lower-quality responses on highly complex or ambiguous queries compared to GPT-4o [19, 20]. This trade-off was acceptable across the pilot evaluation data, but edge cases may exist in broader deployments.

In-memory FAISS indices scale efficiently up to approximately 100 000 candidate profiles. Beyond this threshold, migration to distributed vector databases such as Qdrant or Weaviate would be required [21, 22].

A cold-start challenge exists for new users registering with minimal profile data; match quality improves progressively through conversational onboarding interactions.

The pilot was conducted with 12 Mumbai restaurants over three months. While results are statistically significant, broader validation across diverse hospitality settings (hotels, resorts, event venues), geographies, and languages will be needed to fully characterise generalisability.

8. CONCLUSION AND FUTURE WORK

This paper has introduced an agentic AI framework for semantic workforce matching in the hospitality do-main and demonstrated substantial improvements over tra-ditional recruitment systems [16, 20]. SmartServe achieves 82.4% matching precision — more than double the 38% of keyword-based approaches — while reducing time-to-hire by 74% (12.3 to 3.2 days). A hybrid SLM-LLM architec-ture with role-based model selection reduces overall operational costs by 54% [19, 23]. Employer satisfaction in-creased from 3.2 to 4.6 out of 5, and system response times remain under 4 seconds for complex queries [28]. These results demonstrate that agentic decomposition, semantic similarity modelling, and cost-aware AI design are effec-tive for real-world workforce automation.

Four directions for future work are identified: (1) scaling to 100 000+ profiles through distributed vector databases such as Qdrant or Weaviate [22]; (2) incorporating re-inforcement learning for

continuous improvement based on observed hiring outcomes [17]; (3) extending semantic matching to multilingual embeddings for international and cross-regional deployment [9]; (4) conducting large-scale longitudinal validation across a diverse portfolio of hospitality enterprises to characterise system robustness, fair-ness, and long-term operational impact [25].

9. ACKNOWLEDGMENTS

The authors extend thanks to the participating Mumbai restaurants and hospitality businesses for their cooperation during the pilot programme, and to the reviewers for their constructive feedback.

10. REFERENCES

[1] Pillai, M. and Sivathanu, D. 2020. Adoption of AI-based chatbots for hospitality and tourism. *Int. J. Con-temporary Hospitality Management* 32, 10, 3199–3226.

[2] Tussyadiah, S. and Miller, M. 2021. Perceived im-pacts of artificial intelligence and responses to posi-tive behavior change intervention. *J. Travel Research* 60, 3, 618–637.

[3] Law, R., Li, G., Fong, D.K., and Han, X. 2019.Tourism demand forecasting: A deep learning ap-proach. *Tourism Management* 74, 410–423.

[4] Huang, Y. and Rust, R. 2018. Artificial intelligence in service. *J. Service Research* 21, 2, 155–172.

[5] Sharma, A., Mehta, R., and Patel, K. 2024. AI-driven smart hospitality management systems. *Int. J. Multi-disciplinary Research* 6, 3, 1–15.

[6] Jarrahi, M.H. 2018. Artificial intelligence and the fu-ture of work: Human-AI symbiosis in organizational decision making. *Business Horizons* 61, 4, 577–586.

[7] Pereira, K. and Bavik, A. 2021. The impact of artifi-cial intelligence on workers: Evidence from hospital-ity. *Int. J. Hospitality Management* 98, 103–115.

[8] Prentice, C., Loureiro, X., and Guerreiro, M. 2021. Understanding AI adoption in the hospitality industry. *Int. J. Contemporary Hospitality Management* 33, 11, 3988–4008.

[9] Reimers, N. and Gurevych, I. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. EMNLP*, 3982–3992.

[10] Devlin, J., Chang, M., Lee, K., and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional trans-formers for language understanding. In *Proc. NAACL-HLT*, 4171–4186.

[11] Vaswani, A. et al. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

[12] Liu, Y. et al. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.

[13] Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proc. ICLR*.

[14] Bojanowski, P. et al. 2017. Enriching word vectors with subword information. *Trans. ACL* 5, 135–146.

[15] Peters, M.E. et al. 2018. Deep contextualized word representations. In *Proc. NAACL-HLT*, 2227–2237.

[16] Wooldridge, M. 2009. *An Introduction to MultiAgent Systems*, 2nd ed. John Wiley & Sons, Chichester, UK.

[17] Russell, S.J. and Norvig, P. 2020. *Artificial Intelli-gence: A Modern Approach*, 4th ed. Pearson.

[18] Weiss, G., Ed. 2013. *Multiagent Systems*, 2nd ed. MIT Press, Cambridge, MA.

[19] Chase, J.S. et al. 2023. LangChain: Building applications with LLMs through composability. arXiv:2310.06770.

[20] Wang, L. et al. 2023. A survey on large language model based autonomous agents. arXiv:2308.11432.

[21] Johnson, J., Douze, M., and Jégou, H. 2021. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* 7, 3, 535–547.

[22] Malkov, Y. and Yashunin, D. 2020. Efficient and ro-bust approximate nearest neighbor search using hier-archical navigable small world graphs. *IEEE Trans. PAMI* 42, 4, 824–836.

[23] Lewis, P. et al. 2020. Retrieval-augmented genera-tion for knowledge-intensive NLP tasks. *Advances in NeurIPS*, 9459–9474.

[24] Khattab, O. and Zaharia, M. 2020. ColBERT: Effi-cient and effective passage search via contextualized late interaction over BERT. In *Proc. ACM SIGIR*, 39–48.

[25] Raghavan, M. and Levine, D. 2020. Self-determination, job design and recruitment in gig economy platforms. *Organization Science* 31, 4,

997–1018.

[26] Chalfin, M. et al. 2016. Productivity and selection of human capital with machine learning. *American Eco-nomic Review* 106, 5, 124–127.

[27] Autor, D. 2015. Why are there still so many jobs? The history and future of workplace automation. *J. Eco-nomic Perspectives* 29, 3, 3–30.

[28] Bersin, J. 2024. HR technology disruptions for 2024: The definitive guide. Josh Bersin Company Research.

[29] McKinsey Global Institute. 2021. The future of work after COVID-19. McKinsey & Company Report, February.

[30] Deloitte. 2024. Global Human Capital Trends: The skills-based organization. Deloitte Insights.