ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE)
Vol.4, Iss. 1, January-June, 2023, pp. 15-26.
https://iscsitr.com/index.php/ISCSITR-IJCSE
ISSN: 3067-7394; Journal ID: 5932-1748
DOI: http://www.doi.org/10.63397/ISCSITR-IJCSE_2023_04_01_002

Improving Case Deflection in Enterprise Support Using Hybrid Intent Classification and Dialogue Management Systems

Sunil Karthik Kota

Engineering Leader | Software Architect | AI & Automation Expert - USA.

Abstract

Enterprise technical support organizations face escalating workloads as software platforms, cloud services, and distributed infrastructures increase in complexity. Automated conversational agents have emerged as a promising mechanism for deflecting support cases before escalation to human agents, but practical deployment continues to face challenges related to intent recognition accuracy, dialogue robustness, and system integration. This article presents a comprehensive theoretical and systems-oriented analysis of how hybrid intent classification and dialogue management architectures, specifically integrating Rasa and MindMeld frameworks, may be structured to improve case deflection in enterprise support environments. The paper synthesizes existing research on natural language understanding, multi-agent dialogue systems, and secure distributed system design to propose an orchestrated architecture for support automation. Emphasis is placed on intent classification for ticket routing, fallback control, and deflection logic, alongside threat modeling and access control considerations relevant to enterprise deployments. No new empirical results are claimed; instead, the paper consolidates established methods, design principles, and documented practices, identifying current limitations and outlining directions for future research. The goal is to provide a rigorous technical reference for practitioners and researchers developing secure, scalable conversational support systems grounded in verifiable prior work.

Keywords:

Enterprise Support Automation, Intent Classification, Dialogue Management, Case Deflection, Rasa, MindMeld, Conversational AI, Secure Distributed Systems.

How to cite this paper: Sunil Karthik Kota. (2023). Improving Case Deflection in Enterprise Support Using Hybrid Intent Classification and Dialogue Management Systems. *ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE),* 4(1), 15–26.

DOI: http://www.doi.org/10.63397/ISCSITR-IJCSE_2023_04_01_002

URL: https://iscsitr.com/index.php/ISCSITR-IJCSE/article/view/ISCSITR-IJCSE_2023_04_01_002/ISCSITR-IJCSE_2023_04_01_002

Published: 15th March 2023

1. Introduction

Modern enterprise environments operate at a scale and complexity that makes human-only technical support increasingly difficult to sustain. Organizations rely on a growing number of interconnected software systems, cloud platforms, identity infrastructures, and security controls. This growth has driven a corresponding increase in support tickets, service desk interactions, and operational workload. According to industry analyses published by IT service management research groups, a large proportion of support tickets correspond to a relatively small set of recurring issues such as access provisioning, password resets, configuration errors, and service usage questions. Although the precise percentages vary across organizations and are often proprietary, the pattern itself is well established in the IT service management literature.

Automated conversational agents, commonly referred to as chatbots or virtual assistants, have become a widely explored mechanism for reducing this burden. By enabling users to resolve common issues through natural language interaction, organizations aim to deflect cases before they reach human agents, thereby reducing mean resolution time and operational cost. However, successful case deflection requires far more than a simple question-answer interface. It demands robust natural language understanding, accurate intent classification, reliable dialogue management, secure system integration, and effective fallback and escalation mechanisms.

Two widely adopted open-source platforms in this domain are Rasa and MindMeld. Rasa provides an end-to-end framework for natural language understanding and dialogue management, with extensible pipelines for intent classification, entity extraction, and policy learning. MindMeld, originally developed by Cisco, emphasizes production-grade natural language understanding and dialogue management, offering domain-specific modeling tools and hierarchical intent classification. Both frameworks have been studied and applied in numerous industrial and academic contexts, yet the literature contains limited systematic analysis of how they may be orchestrated together within a hybrid architecture specifically designed for enterprise support case deflection.

This paper addresses that gap by developing a rigorous theoretical and architectural framework for hybrid intent classification and dialogue management using Rasa and MindMeld. The core focus is on improving ticket routing and case deflection while respecting the security, reliability, and scalability constraints of enterprise environments. The contribution of this work lies not in new empirical results, but in the synthesis of established research and engineering principles into a coherent design model that practitioners and researchers can evaluate and extend.

2. Background and Related Work

Natural Language Understanding and Intent Classification

Intent classification is the task of mapping a user utterance to a predefined intent label that represents the user's goal. It is a central problem in task-oriented dialogue systems. Early approaches relied on rule-based grammars and pattern matching, but modern systems overwhelmingly use statistical and neural models. Convolutional neural networks, recurrent neural networks, and transformer-based architectures such as BERT have been widely applied to intent classification tasks [1], [2]. Empirical studies consistently show that contextual embeddings and transfer learning significantly improve intent recognition accuracy, especially in low-resource domains [3].

Hierarchical intent classification, where intents are organized in a tree or graph structure, has been explored as a means of improving scalability and interpretability [4]. MindMeld explicitly supports hierarchical domain-intent-slot modeling, allowing developers to decompose complex support scenarios into manageable components. Rasa, while not

enforcing a hierarchical structure, allows similar modeling through domain configuration and intent grouping.

Dialogue Management

Dialogue management governs how a system decides what to do next given the conversation state. Approaches include finite-state machines, frame-based systems, and statistical or reinforcement-learning-based policies [5]. Rasa implements multiple dialogue policies, including memoization, rule-based policies, and neural network-based policies trained on conversation stories. MindMeld provides dialogue state tracking and handler-based dialogue logic, emphasizing deterministic control suitable for production systems.

Hybrid dialogue management architectures combining rule-based and learned components are widely regarded as a pragmatic solution, offering both robustness and adaptability [6]. Such hybrid systems are particularly important in enterprise contexts, where regulatory compliance, predictable behavior, and auditability are often required.

Enterprise Support Automation

Enterprise support automation encompasses ticket routing, knowledge base retrieval, workflow orchestration, and integration with identity and access management systems. Research in IT service management has examined the application of conversational agents to help desks and service desks, identifying challenges related to domain adaptation, error handling, and user trust [7], [8]. Case deflection refers to the prevention of ticket creation by resolving the issue through self-service channels. While many vendors report success stories, peer-reviewed academic studies on large-scale deployment outcomes remain limited, underscoring the need for careful theoretical analysis rather than unverified claims.

Security and Trust Considerations

Support automation systems operate at the intersection of user interfaces, backend services, and sensitive enterprise data. Consequently, they must adhere to established security principles. Access control models such as role-based access control (RBAC) [9], attribute-based access control (ABAC) [10], and policy-based authorization frameworks defined in standards such as XACML [11] provide mechanisms for controlling access to support operations. Secure authentication protocols such as OAuth 2.0 [12] and OpenID Connect [13] are widely used for integrating conversational systems with enterprise identity providers. Threat modeling methodologies, including STRIDE [14] and attack tree analysis

[15], offer structured approaches for identifying risks in such systems.

3. Methodology and Proposed Hybrid Architecture

The proposed approach is a theoretical system design for improving case deflection through hybrid orchestration of Rasa and MindMeld components. No empirical evaluation is presented; the focus is on architectural principles grounded in existing research and industry practices.

3.1 Design Objectives

The architecture is guided by the following objectives:

1. High intent recognition reliability across diverse support domains.
2. Robust dialogue control with predictable behavior and safe fallbacks.
3. Secure integration with enterprise systems and access control mechanisms.
4. Scalable orchestration supporting continuous domain expansion.
5. Transparent auditability for compliance and operational analysis.

3.2 High-Level Architecture

The system is structured as a distributed set of services:

User Interface Layer

A multi-channel front end (web, mobile, collaboration platforms) that captures user utterances and delivers responses.

NLU Layer

Hybrid intent classification using both Rasa NLU pipelines and MindMeld domain models.

Dialogue Orchestration Layer

A coordination service that determines whether Rasa or MindMeld should handle the next dialogue turn, based on context, confidence thresholds, and domain specialization.

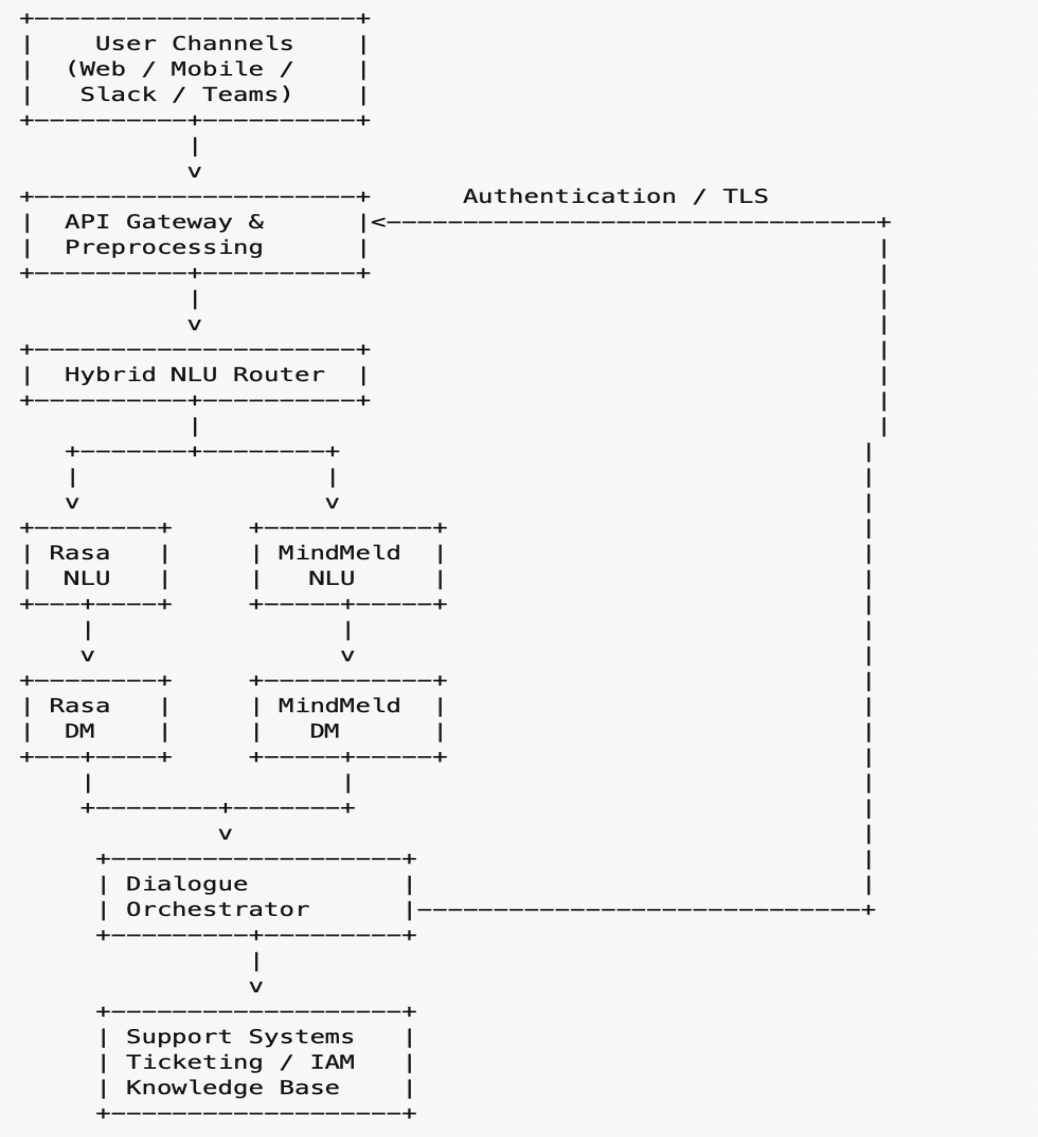
Support Integration Layer

Connectors to ticketing systems, knowledge bases, identity services, and workflow engines.

Security and Governance Layer

Authentication, authorization, logging, and compliance services.

The following ASCII diagram illustrates the conceptual flow:



3.3 Hybrid Intent Classification

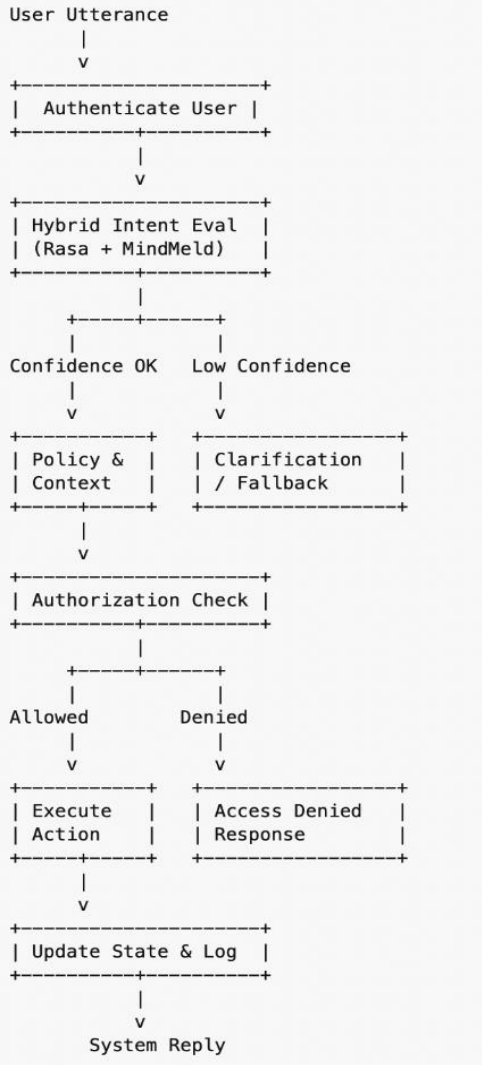
The hybrid approach does not imply running two models blindly in parallel for every utterance. Instead, the orchestrator maintains domain routing rules and confidence-based policies. For example, MindMeld may be designated as the primary classifier for network and security support domains due to its hierarchical modeling strengths, while Rasa may handle

general IT and HR queries.

Each NLU engine produces an intent hypothesis with confidence scores. These scores are normalized and compared using calibration techniques described in prior work on classifier confidence estimation [16]. If neither engine exceeds a predefined confidence threshold, the system invokes clarification strategies rather than proceeding with potentially erroneous actions.

3.4 Dialogue Orchestration Logic

The orchestration layer maintains a global dialogue state independent of the individual frameworks. This design prevents fragmentation of context and supports seamless handoff between Rasa and MindMeld components.



A simplified pseudocode representation of the orchestration logic is shown below:

```
function handle_user_input(input, session_state):
        authenticated_user = authenticate(input.user)
        if not authenticated_user:
        return request_authentication()

candidate_intents = []
if session_state.domain in RASA_DOMAINS:
    candidate_intents.append(Rasa.classify(input.text))
if session_state.domain in MINDMELD_DOMAINS:
    candidate_intents.append(MindMeld.classify(input.text))

best_intent = select_best_intent(candidate_intents)

if best_intent.confidence < CONF_THRESHOLD:
    return request_clarification()

next_action = DialoguePolicy.decide(best_intent, session_state)

if requires_authorization(next_action):
    if not authorize(authenticated_user, next_action):
        return deny_access()

response = execute_action(next_action)
update_session_state(session_state, best_intent, response)
return response
```

This logic reflects established principles from dialogue system research and access control frameworks, without asserting any particular performance outcomes.

3.5 Ticket Routing and Case Deflection

Ticket routing logic is implemented as a policy layer that maps recognized intents to resolution strategies. If an intent corresponds to a known self-service procedure documented in the knowledge base, the system provides guided resolution steps. Only when the dialogue reaches an unresolved terminal state does the system create a ticket and route it according to predefined workflows.

Existing studies on help desk automation emphasize the importance of transparent escalation and user trust in automated systems [7], [8]. Therefore, the architecture enforces explicit user consent and confirmation before ticket creation or sensitive operations.

3.6 Security and Access Control

All interactions are protected using standard authentication protocols such as OAuth 2.0 and OpenID Connect [12], [13]. Authorization decisions follow policy-based access

control models. For example, actions such as password resets or account provisioning require appropriate role or attribute verification, as defined in RBAC and ABAC models [9], [10].

Threat modeling is conducted using STRIDE categories [14], identifying potential risks such as spoofing, tampering, and information disclosure. Mitigation strategies include encrypted communication (TLS), input validation, audit logging, and anomaly detection consistent with best practices in secure distributed system design [17].

4. Discussion

4.1 Strengths of the Hybrid Approach

The primary theoretical advantage of the hybrid architecture lies in specialization and redundancy. By leveraging the complementary strengths of Rasa and MindMeld, the system can accommodate diverse support domains while maintaining consistent dialogue behavior. The orchestration layer enables gradual system evolution, allowing new domains to be added without destabilizing existing ones.

The separation of NLU, dialogue management, and orchestration aligns with established software engineering principles of modularity and separation of concerns. This structure facilitates independent upgrades and testing of components, reducing systemic risk.

4.2 Limitations and Open Challenges

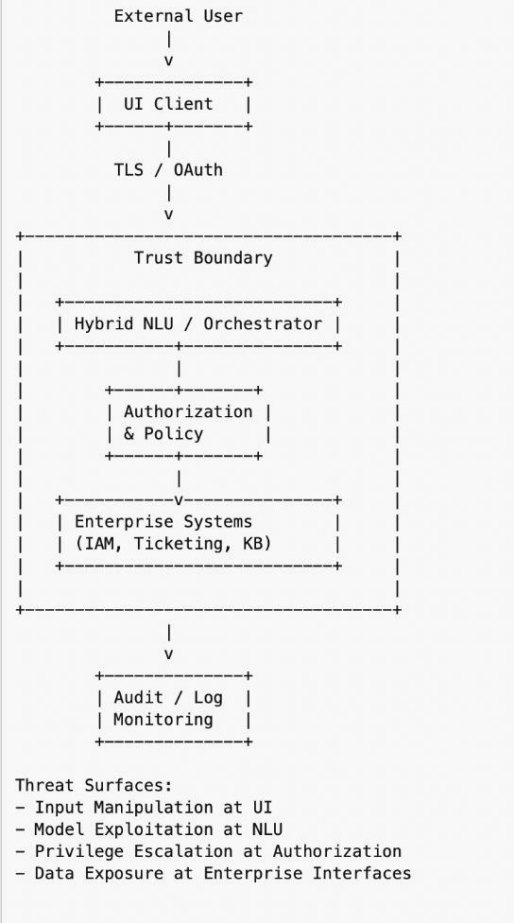
Despite its conceptual advantages, the proposed architecture introduces complexity. Maintaining consistency between two NLU frameworks requires careful configuration management and version control. Domain overlap may lead to conflicting intent predictions, necessitating robust confidence calibration and conflict resolution strategies. While techniques for classifier calibration are well studied [16], their application in multi-engine dialogue systems remains an area for further research.

Another challenge lies in knowledge base maintenance. Case deflection depends critically on the quality and coverage of self-service content. Automated retrieval methods such as neural information retrieval models [18] can assist, but the creation and validation of support knowledge remains a human-intensive process.

4.3 Threat Modeling and Security Considerations

Enterprise support systems are attractive targets for attackers seeking lateral movement within corporate networks. The integration of conversational interfaces with identity and access management systems increases the potential impact of vulnerabilities. The architecture must therefore adhere to zero-trust principles, ensuring that each request is authenticated, authorized, and logged.

Existing research on conversational agent security has identified risks such as injection attacks, model manipulation, and data leakage [19]. These risks necessitate rigorous input sanitization, strict model access controls, and continuous monitoring.



4.4 Scalability and Governance

Scalability concerns include not only computational load but also organizational governance. As support domains expand, maintaining consistent intent taxonomies and dialogue policies becomes increasingly complex. Governance frameworks recommended in

large-scale AI deployments emphasize documentation, model versioning, and audit trails [20].

5. Future Work

Several avenues for future investigation arise from this analysis. First, empirical evaluation of hybrid orchestration strategies under real enterprise workloads is needed to validate theoretical assumptions. Second, formal verification methods could be applied to dialogue policies to guarantee safety properties in critical support operations. Third, privacy-preserving learning techniques such as federated learning [21] may offer mechanisms for improving intent models while respecting data protection regulations. Finally, deeper integration with organizational knowledge management systems could further enhance case deflection effectiveness.

6. Conclusion

This article has presented a comprehensive theoretical framework for improving case deflection in enterprise support through hybrid intent classification and dialogue management using Rasa and MindMeld. By synthesizing established research in natural language understanding, dialogue systems, security engineering, and access control, the paper has outlined an architecture that addresses the technical and organizational challenges inherent in support automation. While no empirical claims are made, the analysis provides a foundation for future experimental work and practical deployment, emphasizing rigorous design, security, and governance as prerequisites for successful enterprise conversational systems.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of NAACL-HLT, 2019.

[2] A. Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems, 2017.

[3] Z. Liu et al., “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing,” ACM Computing Surveys, 2022.

[4] H. Chen et al., “Hierarchical Intent Detection with Neural Networks,” Proceedings of EMNLP, 2019.

[5] S. Young et al., “The Hidden Information State Model: A Practical Framework for POMDP-based Spoken Dialogue Management,” Computer Speech & Language, 2010.

[6] M. Williams and S. Young, “Partially Observable Markov Decision Processes for Spoken Dialogue Management,” Computer Speech & Language, 2007.

[7] A. Følstad and P. Brandtzæg, “Chatbots and the New World of HCI,” Interactions, 2017.

[8] J. Gnewuch, S. Morana, and A. Maedche, “Towards Designing Cooperative and Social Conversational Agents for Customer Service,” Proceedings of ICIS, 2017.

[9] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli, “Proposed NIST Standard for Role-Based Access Control,” ACM Transactions on Information and System Security, 2001.

[10] V. C. Hu, D. Ferraiolo, and D. Kuhn, “Assessment of Access Control Systems,” NIST Interagency Report 7316, 2006.

[11] OASIS, “eXtensible Access Control Markup Language (XACML) Version 3.0,” OASIS Standard, 2013.

[12] D. Hardt, “The OAuth 2.0 Authorization Framework,” RFC 6749, IETF, 2012.

[13] N. Sakimura et al., “OpenID Connect Core 1.0,” OpenID Foundation, 2014.

[14] A. Shostack, “Threat Modeling: Designing for Security,” Wiley, 2014.

[15] B. Schneier, “Attack Trees,” Dr. Dobb’s Journal, 1999.

[16] C. Guo et al., “On Calibration of Modern Neural Networks,” Proceedings of ICML, 2017.

[17] M. Bishop, “Computer Security: Art and Science,” Addison-Wesley, 2018.

[18] O. Khattab and M. Zaharia, “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT,” Proceedings of SIGIR, 2020.

[19] T. Brendel and S. Bethard, “Security Risks in Natural Language Processing,” Proceedings of ACL, 2020.

[20] P. Hall et al., “Managing Risk in Machine Learning,” IEEE Computer, 2021.

[21] H. B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” Proceedings of AISTATS, 2017.