

Knowledge or Action? Automation Boundary Prediction with Intent Discovery and Knowledge Use-Case Enablement for Agentic Enterprise Support

Supplementary Material

8. Appendix A: Hierarchical Intent Taxonomy Induction

This phase constructs the hierarchical intent taxonomy used throughout the framework. The taxonomy is induced from raw enterprise support tickets through an iterative process consisting of intent extraction, label normalization, and hierarchical organization. The following prompt templates were used during these stages.

8.1. Initial Intent Extraction Prompt Objective

Extract candidate intent labels from raw enterprise HR support tickets without restricting the model to predefined categories. This step captures the natural diversity of language used by employees when describing support issues.

Input: Each input consists of: (1) Employee-written ticket description, (2) Optional screenshot or log content
Example input: *"TICKET DESCRIPTION: I submitted my HRA declaration but it was rejected in the tax portal. What should I do?"*

Taxonomy Induction Prompt

You are an expert HR Support Analyst and Enterprise Process Taxonomy Specialist. You specialize in analyzing employee HR support tickets and identifying the underlying intent structure of each request. Your task is to extract the hierarchical intent structure of each ticket. The goal is to identify the underlying HR topic and the specific issue the employee is experiencing. Tickets may include both textual descriptions and visual artifacts (such as screenshots of HR portals, payroll pages, or system errors). When available, incorporate the visual context to better understand the employee's issue.

Ticket Content

Employee Description:
<<<TICKET_DESCRIPTION>>>
Attachment Context (if available):
<<<ATTACHMENT_CONTEXT>>>

For the given ticket, extract the

following intent attributes:

1. **Category:** The highest-level HR operational domain which represents the broad HR service area to which the ticket belongs.
2. **Topic:** The functional subject area within the category. Avoid repeating the category name as the Topic should narrow the problem area within it's category
3. **Component** The specific HR element, document, system feature, or operational unit referenced in the request.
4. **Reason** The core action, issue, or request expressed by the employee. Summarize this in a concise phrase (1-3 words).

Guidelines:

- Carefully analyze the entire ticket context.
- If screenshots or logs are available, incorporate their information.
- Do not restrict outputs to predefined labels.
- Use natural terminology that best reflects the employee request.
- Avoid overly generic labels when more specific components are identifiable.
- The reason should represent the core issue or action requested (e.g., "access issue", "declaration rejected", "balance inquiry").

Return the result in JSON format:

```
{ "category": "",  
  "topic": "",  
  "component": "",  
  "reason": "" }
```

Example Output

```
{  
  "category": "Income Tax",  
  "topic": "Tax Declaration",  
  "component": "House Rent Allowance",  
  "reason": "HRA rejection"  
}
```

This stage generates a large set of candidate labels, often containing surface-form variations for the same concept.

8.2. Taxonomy Normalization Prompt

Consolidate semantically equivalent labels extracted during the previous stage. This step reduces label fragmentation caused by variations in terminology.

Input: A list of candidate labels extracted from support tickets. Example input: Salary Slip, Pay Stub, Salary Payslip, Compensation Payroll, Salary Payroll

Taxonomy Normalization Prompt

You are designing a standardized taxonomy for enterprise HR support tickets. You are given a list of labels extracted from employee support tickets. Because employees use diverse language when describing issues, many labels refer to the same concept but appear with different wording. Your task is to consolidate semantically equivalent labels into canonical taxonomy terms.

Candidate Labels

<<<LABELLIST>>>

Instructions:

1. Identify labels that refer to the same underlying HR concept.
2. Group such labels together under a single canonical term.
3. The canonical label should be the most clear and commonly used HR terminology.
4. Preserve semantic meaning while removing redundant wording variations.
5. Avoid merging labels that represent genuinely different HR concepts.
6. Prefer concise canonical labels (1-3 words where possible).

Return the result as a JSON mapping where each canonical label contains the list of equivalent variants.

Output format:

```
"Canonical Label": ["Variant 1",  
"Variant 2", "Variant 3"]
```

Example Output

```
{  
  "Payslip": ["Salary Slip", "Pay Stub",  
  "Salary Payslip"],  
  "Payroll": ["Compensation", "Salary  
  Payroll"]  
}
```

This stage produces a set of normalized taxonomy nodes.

8.3. Hierarchical Taxonomy Construction Prompt

Organize the normalized labels into a hierarchical taxonomy suitable for structured intent classification.

Input: Normalized labels obtained from the previous stage.

Example input: Payroll, Payslip, Tax Declaration, HRA, Section 80C, PF Withdrawal, PF Contribution

Hierarchical Taxonomy Construction Prompt

You are designing a hierarchical taxonomy for enterprise HR support tickets submitted by employees through an internal helpdesk system. You are given a list of normalized intent labels extracted from support tickets. Your task is to organize these labels into a hierarchical structure using the following levels:
Category → Topic → Component

Normalized Labels

<<<NORMALIZED_LABELS>>>

Hierarchy Definitions:

Category: The highest-level HR operational domain that represents a broad area of employee support (e.g., Payroll, Benefits, Retirals, System Access, Leave Management).

Topic: A functional sub-area within a category representing a specific HR process or policy area.

Component: The most granular level representing a specific HR element, document, system feature, or operational unit referenced in support tickets.

Guidelines:

- Ensure each component belongs to only one topic.
- Ensure each topic belongs to only one category.
- Avoid overlapping or duplicate hierarchy nodes.
- Group conceptually related labels together under the same topic.
- Prefer concise and clear HR terminology for category and topic names.
- Maintain logical consistency across the hierarchy.

Return the taxonomy in nested JSON format using the structure:

```
{  
  "Category": {
```

```

    "Topic":
    [
        "Component",
        "Component"
    ]
  }}

```

Example Output

```

{
  "Payroll": {
    "Tax Declaration": [
      "House Rent Allowance",
      "Section 80C"
    ],
    "Payslip": [
      "Payslip Access",
      "Payslip Correction"
    ]
  },
  "Retirals": {
    "Provident Fund": [
      "PF Withdrawal",
      "PF Contribution"
    ]
  }
}

```

The resulting hierarchy defines the validated taxonomy T^* used in subsequent phases of the framework. Each ticket can subsequently be mapped to a Category–Topic–Component tuple within this structure.

9. Appendix B: Taxonomy-Constrained Classification Prompt

Classify enterprise HR support tickets into a validated hierarchical intent taxonomy induced during Phase 1. Unlike open-vocabulary extraction, this stage constrains predictions to predefined taxonomy nodes to ensure consistent labeling across the dataset.

Input: Each prompt instance processes a single support ticket containing:

- Employee-written description
- Optional screenshot attachment
- Optional system log content
- Validated taxonomy T^*

Example ticket input: *"Description: I cannot download my payslip for March."*

Attachment (optional): Screenshot of Workday portal showing "Payslip not available".

Taxonomy Context

The full validated taxonomy T^* is provided to the model as a structured JSON object.

Example (partial):

```

{
  "Salary/Payslip": {
    "Payslip & Documents": [
      "Salary Slip / Payslip",
      "Official Documents"
    ],
    "Salary Components & Clarifications": [
      "Base Pay",
      "Allowances",
      "Salary deduction"
    ]
  }
}

```

During inference, the model must select values only from the provided taxonomy nodes.

Taxonomy-Constrained Classification Prompt

You are analyzing enterprise HR support tickets submitted by employees through an internal helpdesk system. Your task is to classify the ticket into a hierarchical intent taxonomy that has already been validated by HR domain experts.

The taxonomy structure follows the hierarchy:

Category → Topic → Component

You must select Category, Topic, and Component strictly from the taxonomy provided below.

Validated Taxonomy

<<<TAXONOMY_JSON>>>

Ticket Content

Employee Description:

<<<TICKET_DESCRIPTION>>>

Attachment Context (if available):

<<<ATTACHMENT_CONTEXT>>>

Instructions:

- Carefully analyze the ticket description and attachment context.
 - Select the most appropriate Category, Topic, and Component from the taxonomy.
 - Do not generate new labels outside the provided taxonomy.
 - Identify the core issue expressed in the ticket and summarize it as a short phrase (1{3 words).
- Return the output in JSON format:

```
{ "category": "",
  "topic": "",
  "component": "",
  "reason": "" }
```

Example Output

```
{ "category": "Salary/Payslip",
  "topic": "Payslip & Documents",
  "component": "Salary Slip / Payslip",
  "reason": "payslip access" }
```

10. Appendix C: Automation Boundary Prediction Prompt

Predict the automation boundary of enterprise HR support tickets by classifying them into two resolution tiers:

- **T0 (Knowledge-Resolvable):** The request can be resolved using policy documentation, procedural guidance, or knowledge articles without accessing employee-specific data.
- **T1 (System-Dependent):** Resolution requires access to employee-specific records, system workflows, or external enterprise tools.

This classification determines whether a ticket can be handled through knowledge-based automation or requires agent intervention and system integration.

Input: Each prompt instance processes a single support ticket with contextual information obtained from the previous classification stage.

Inputs include:

(Category → Topic → Component)

- Employee-written ticket description
- Optional screenshot or system log context
- Optional system log content
- Hierarchical intent labels extracted in Phase 2

Example ticket input:

Ticket Description: "I cannot download my payslip for March."

Attachment Context (optional): Screenshot of Workday portal showing "Payslip not available". *Category: Salary/Payslip Topic: Payslip & Documents Component: Salary Slip / Payslip*

Ticket Description: "I cannot download my payslip for March."

Attachment Context (optional): Screenshot of Workday portal showing "Payslip not available".

Automation Boundary Prediction Prompt

You are analyzing enterprise HR support tickets to determine whether the request can be resolved through knowledge-based documentation or requires system-level intervention.

Your task is to classify the ticket into one of two automation tiers:

T0 (Knowledge-Resolvable) The ticket can be resolved using policy explanations, procedural documentation, or general guidance without accessing employee-specific data.

T1 (System-Dependent) Resolution requires retrieving employee-specific data, checking payroll records, verifying system workflows, or interacting with enterprise systems.

Ticket Context

Category: <<<CATEGORY>>> Topic: <<<TOPIC>>> Component: <<<COMPONENT>>> Employee Description: <<<TICKET_DESCRIPTION>>> Attachment Context (if available): <<<ATTACHMENT_CONTEXT>>>

Reason through the following questions:

- 1. Does resolving this request require employee-specific data (e.g., payroll records, leave balances, personal account details)?*
- 2. Can the issue be answered using general documentation or policy guidance?*
- 3. Does resolving the issue require system access or workflow checks?*

If the request can be answered using general knowledge or documentation, classify it as T0.

If the request requires checking employee-specific information or interacting with enterprise systems, classify it as T1.

Return the result in JSON format:

```
{
  "automation_scope": "T0 or T1",
}
```

Example Output

```
{
  "automation_scope": "T0",
  "reason": "general guidance request"
}
```

11. Appendix D: Knowledge Use-Case Extraction Prompt

Transform clusters of knowledge-resolvable (T0) support tickets into structured knowledge specifications that can

guide subject matter experts in authoring knowledge articles. The goal is to identify recurring employee intent patterns and convert them into reusable knowledge use-cases.

Input: Each prompt instance processes a cluster of T0 tickets grouped by Component or Topic within the validated taxonomy.

Inputs include:

- Taxonomy context (Category → Topic → Component)
- Aggregated ticket descriptions belonging to the same cluster
- Optional screenshot or attachment context

Example ticket input:

Clustered Ticket Descriptions (samples):

1. "My HRA declaration was rejected. What should I do?"

2. "Why is HRA not reflecting in my tax calculation?"

3. "How do I submit documents for HRA proof?"

4. "Where do I upload HRA rent receipts?" Category:

Income Tax Topic: Tax Deductions & Declarations Component: House Rent Allowance (HRA)

Clustered Ticket Descriptions (samples):

1. "My HRA declaration was rejected. What should I do?"

2. "Why is HRA not reflecting in my tax calculation?"

3. "How do I submit documents for HRA proof?"

4. "Where do I upload HRA rent receipts?"

Knowledge Use-Case Extraction Prompt

You are analyzing clusters of enterprise HR support tickets to identify recurring employee information needs that can be resolved through knowledge articles. The tickets in this cluster have already been classified as knowledge-resolvable (T0), meaning they can be answered through documentation or procedural guidance.

Taxonomy Context

Category: <<<CATEGORY>>> Topic:
<<<TOPIC>>> Component: <<<COMPONENT>>>

Clustered Ticket Descriptions

<<<TICKET_CLUSTER>>>

Your task is to identify common intent patterns within these tickets and convert them into structured knowledge use-cases.

For the given cluster:

1. Identify distinct recurring employee questions or issues.

2. Group similar tickets into a single use-case when they represent the same underlying intent.
 3. Estimate how frequently each use-case appears in the cluster.
 4. Generate canonical question formulations that represent how employees typically ask about this issue.
 5. Provide a concise description summarizing the employee objective behind each use-case.
- Return the result in JSON format:

```
{
  "taxonomy_path": {
    "category": "",
    "topic": "",
    "component": ""
  },
  "use_cases": [
    {
      "use_case_title": "",
      "ticket_count": "",
      "summary": "",
      "canonical_questions": [
        "",
        ""
      ]
    }
  ]
}
```

Example Output

```
{
  "taxonomy_path": {
    "category": "Income Tax",
    "topic": "Tax Deductions &
      Declarations",
    "component": "House Rent
      Allowance (HRA)"
  },
  "use_cases": [
    {
      "use_case_title": "HRA
        declaration rejection",
      "ticket_count": 12,
      "summary": "Employees need
        guidance when HRA claims are
        rejected due to missing documents
        or incorrect submission.",
      "canonical_questions": [
        "Why was my HRA declaration
        rejected?",
        "How do I fix an HRA claim
        rejection?"
      ]
    }
  ]
}
```