Nova n8n Workflow Analysis

Executive Summary

The Nova system is an intelligent prompt routing architecture that analyzes user queries and directs them to specialized agent workflows based on content classification. The system consists of a main workflow that handles prompt categorization and routing, along with specialized agent workflows for different types of tasks.

System Architecture Overview

Show Image

The Nova system consists of two main components:

1. **Nova Main Workflow** - An intelligent router that categorizes prompts and distributes them to specialized agents
2. **Agent Workflows** - Specialized processing workflows for different types of queries (Emil, Ivan, Lola, and Nova)

Nova Main Workflow

Entry Points

The workflow has two entry points:

- **Start Node** - For internal execution
- **Nova Webhook** - For external API triggers

Input Processing Flow

1. **Input Preparation**
   - `Prepare Internal Input` - Formats internal inputs with timestamp and session ID
   - `Extract Prompt` - Handles webhook inputs by extracting prompt data
   - `Merge Inputs` - Combines inputs from both sources
2. **Intent Processing**
   - `Split Multi-Intent Prompts` - Breaks complex prompts into segments based on conjunctions and sentence boundaries
   - Segments are filtered to remove noise and ensure meaningful processing

3. **Prompt Categorization**
   - The `Categorize Prompt` node analyzes prompt text using keyword detection:

Category	Keywords	Agent Assignment
Energy Models	"model", "energy"	Emil
Reports	"report", "summary", "write"	Lola
Visualizations	"chart", "image", "visual"	Ivan
Code	"script", "python"	Ivan
Math	Mathematical expressions (regex detected)	Ivan
General	All other queries	Nova

4. **Agent Routing**
   - The `Route to Agent` node directs the flow to specialized workflows:
     - **Route 0**: Emil (energy modeling specialist)
     - **Route 1**: Ivan (visualization and code specialist)
     - **Route 2**: Lola (report generation specialist)
     - **Route 3**: Nova (general purpose assistant)
5. **Workflow Execution**
   - Each agent has a dedicated workflow execution node:
     - `Execute Emil` - Calls the Emil workflow
     - `Execute Ivan` - Calls the Ivan workflow
     - `Execute Lola` - Calls the Lola workflow
   - Nova's implementation is directly in the main workflow

Nova Implementation

For queries routed to Nova:

1. **Input Preparation**
   - `Nova Prepare Input` - Ensures all required fields exist (prompt, history, sessionId, timestamp)
2. **LLM Processing**
   - `Nova OpenAI Answer` - Uses GPT-4.1-nano model with the system prompt:

     ```
     You are Nova. Answer the user's question clearly and helpfully.
     ```
3. **Output Formatting**
   - `Nova Format Output` - Formats the LLM response with metadata including agent attribution

Result Handling

All agent responses pass through dedicated result handlers:

- `Emil Result Handler`
- `Ivan Result Handler`
- `Lola Result Handler`
- `Nova Result Handler`

These handlers standardize the response format with:

- Agent name
- Response content
- Original prompt
- Timestamp
- Session ID

The standardized response is then returned through the `Return Agent Response` node.

Prompt Testing Workflow

A separate workflow exists for testing the Nova system:

1. **Trigger and Input**
   - `Manual Trigger` - Starts the test process
   - `Set Prompt` - Contains the test query: "Build a solar model for spain and what is 2 + 2"
2. **Conversation Management**
   - `Code` - Creates/updates conversation history
   - `Execute Workflow` - Calls the Nova workflow
   - `Add Assistant Reply` - Appends the response to conversation history

Workflow Logic Highlights

Multi-Intent Detection

The system intelligently splits complex prompts with multiple questions or statements:

```javascript
// Split on conjunctions, questions, or sentence endings
const segments = prompt
  .split(/(?:\band\b|\balso\b|[.?!;])+/i)
  .map(s => s.trim())
  .filter(s => s.length > 3); // Filter out noise
```

In the example prompt "Build a solar model for spain and what is 2 + 2":

- It would split into: ["Build a solar model for spain", "what is 2 + 2"]
- The first segment would route to Emil (energy model specialist)
- The second segment would route to Ivan (math specialist)

Prompt Classification Logic

```javascript
let category = "general";
if (prompt.includes("model") || prompt.includes("energy")) {
  category = "energy_model";
} else if (prompt.includes("report") || prompt.includes("summary") || prompt.includes("write"))
  category = "report";
} else if (prompt.includes("chart") || prompt.includes("image") || prompt.includes("visual")) {
  category = "chart";
} else if (prompt.includes("script") || prompt.includes("python")) {
  category = "code";
} else if (/\d+\s*[+\-*/]\s*\d+/.test(prompt)) {
  category = "math";
}

let agent = "Nova";
if (category === "energy_model") agent = "Emil";
else if (category === "report") agent = "Lola";
else if (category === "chart" || category === "code") agent = "Ivan";
```

System Benefits

1. **Specialized Processing**: Each agent is optimized for specific types of queries
2. **Multi-Intent Handling**: Complex queries with multiple questions are appropriately routed
3. **Consistent Response Format**: Standardized output structure regardless of processing agent
4. **Session Persistence**: Conversations maintain history through session IDs
5. **Extensibility**: New agents and categories can be added by extending the classification logic

Implementation Considerations

1. **Agent Workflows**: The main workflow references external workflows for Emil, Ivan, and Lola
2. **LLM Integration**: Uses OpenAI's GPT-4.1-nano for the Nova agent
3. **Webhook Integration**: External systems can trigger the workflow via webhook

Conclusion

The Nova system demonstrates an efficient approach to prompt routing using n8n's workflow capabilities. By categorizing queries and directing them to specialized agents, the system can provide optimized responses for different types of requests while maintaining a unified interface.