Invoice Reconciliation Multi-Agent System

Written Analysis

1. Where does OCR or extraction fail in your system, and how do the agents compensate?

Answer: The most fragile part of the system is document understanding and OCR, which is expected in any real-world invoice processing pipeline. Tesseract OCR struggles with low-quality scans, skewed or rotated documents, stamps, watermarks, and complex or poorly aligned tables. In addition, the LLM-based extraction step sometimes returns invalid JSON, misses fields, or produces inconsistent formatting. These are not rare edge cases; they are common in production document workflows.

In our implementation, these weaknesses are handled in a defensive and transparent way. The Document Agent uses safe JSON parsing and fallback logic. If the LLM output cannot be parsed, the system creates an empty but valid invoice structure instead of crashing, so the pipeline never breaks. Downstream agents are designed to detect missing or weak data. If no items or key fields are found, the Matching and Discrepancy Agents treat the case as unreliable, and the Resolution Agent escalates it to a human reviewer. The system treats low-quality extraction as uncertainty rather than trying to guess, which mirrors real accounting workflows.

2. How would you improve accuracy from about 70% to 95%?

Answer: Several improvements would significantly increase accuracy. The most important change would be replacing basic OCR with a layout-aware document understanding system such as Azure Form Recognizer, Google Document AI, or Amazon Textract, which understand tables and key-value pairs much better.

The extraction step should also move to schema-guided structured outputs or function calling, with strict validation and retry logic. For matching, the current fuzzy string matching works but is limited. Using semantic embeddings for item matching would reduce both false positives and false negatives. A human feedback loop could be added, where corrections made by reviewers are stored and reused. Finally, maintaining vendor-specific templates or profiles would improve accuracy, since most invoices come from repeat suppliers.

3. How would you validate and scale this system to 10,000 invoices per day?

Answer: At that scale, the system would need to become asynchronous and distributed. Each agent should run as a worker behind a queue, allowing OCR, extraction, and matching to run in parallel across many documents. This enables horizontal scaling and better use of compute resources.

Strong monitoring would be essential. The system should track extraction success rate, auto-approval rate, escalation rate, and human correction rate. A labeled evaluation dataset should be run continuously to detect regressions. Cost and latency would be controlled by caching known vendors and only calling the LLM when necessary.