SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Reliable Multi-Agent Systems for Automated Technical Data Acquisition and Validation

Yange Zheng

SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Reliable Multi-Agent Systems for Automated Technical Data Acquisition and Validation

Zuverlässige Multi-Agenten-Systeme für die automatisierte technische Datenerfassung und -validierung

Author:	Yange Zheng
Examiner:	Prof. Dr. Viktor Leis
Advisor:	Dr. Alexander Schiffmacher
Supervisor:	Joe Yu
Submission Date:	01.07.2025

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 01.07.2025 Yange Zheng

Acknowledgments

I owe special thanks to **Dr. Alexander Schiffmacher** (BMW AG), whose guides thesis from start to finish, with invaluable insights and support throughout the process.

I would also like to thank **Joe Yu** (BMW AG & TUM) for providing timely feedback and encouragement on key milestones.

Many thanks to **Stefan Merbele** (BMW AG) for his generous support with data access and infrastructure.

...

...

Finally, I am grateful to my family and friends for their unwavering support and encouragement during this journey.

Munich, July 2025 Yange Zheng

Abstract

In industrial procurement, engineers often work with **Quotation Analysis Forms (QAFs)** provided by suppliers—semi-structured documents containing part descriptions, electrical specifications, and inconsistent identifiers. While Manufacturer Part Numbers (MPNs) are typically present, they are often embedded in noisy text alongside other attributes, making reliable extraction and interpretation difficult. Automating the full pipeline—from QAF to datasheet to validated component specifications—requires coordinating diverse tools: language models, web search, document parsing, and schema validation. Rigid scripts and monolithic workflows tend to break in the face of data variation or missing fields. This thesis addresses that gap by designing a **modular multi-agent system** where each agent specializes in one task (e.g., MPN extraction, datasheet retrieval, VLM parsing), enabling flexible, fault-tolerant, and interpretable orchestration across real-world data acquisition scenarios.

Contents

1 Introduction

Example QAF Line (synthetic)

```
10UF 25V X5R 0603 C1608X5R1E106K080AC RoHS Reel 4000pcs $0.073
```

Figure 1.1: Typical supplier-submitted QAF line (synthetic example).

1.1 Background

The digitalisation of automotive supply chains has accelerated in recent years, yet many data-critical workflows still depend on semi-structured documents and manual interpretation. A prominent example at BMW is the *Quotation Analysis Form* (QAF), a file exchanged during supplier negotiations that lists part designations, preliminary prices, delivery conditions, and essential technical details. Although QAFs frequently include an identifier resembling a Manufacturer Part Number (MPN), suppliers often embed the MPN within longer descriptive strings—alongside electrical characteristics and ordering information—or omit it altogether. As shown in Figure 1.1, QAF lines typically appear as unstructured, single-line entries that conflate multiple attributes in free-form text. Consequently, engineers must manually inspect each line, identify or correct the MPN, locate the corresponding datasheet, and extract key specifications—such as maximum voltage or operating temperature—for input into BMW's internal cost and qualification systems. With several thousand parts assessed per program phase, this *ad hoc* process consumes hundreds of engineer-hours and remains susceptible to transcription errors.

1.2 Problem Statement

Previous attempts to automate QAF processing primarily relied on regular expressions to extract Manufacturer Part Numbers (MPNs) from raw supplier text. While effective in narrowly defined cases, these approaches were brittle—minor variations in formatting, inconsistent delimiters, or embedded product descriptions frequently caused extraction to fail. A further limitation was ambiguity: when the regular expression returned a null result, it was unclear whether no MPN was present or if the pattern simply failed to match a valid one. To address these issues, large language models (LLMs) offer a more robust alternative. By reasoning over free-form text and recognizing contextual cues, LLMs can identify MPNs even when they are embedded in descriptive strings or surrounded by supplier-specific notations.

1.3 Research Objectives

The central objective of this thesis is to design, implement, and evaluate a **reliable multi-agent system** that transforms noisy QAF inputs into validated, structured component specifications. To operationalise this goal, the work addresses four research questions:

RQ1. How accurately can a language-model agent extract MPNs from noisy, multilingual part designation fields in semi-structured QAFs?

RQ2. How reliably can cooperative retrieval agents—using distributor APIs and parse datasheet to complete the technical parameters?

RQ3. How can conflicting or partial parameter values from distributors and datasheets be aligned and fused into a consistent, validated representation?

RQ4. How can provenance be captured and maintained at each stage to ensure full traceability of the extracted data?

1.4 Proposed Approach

To answer these questions, the thesis proposes a modular pipeline whose agents each address a single sub-task:

1. An **MPN Extraction Agent** uses GPT-4 with prompt engineering and regex to isolate MPN from QAF text.
2. Two **Retrieval Agents** operate in parallel:
   (i) An *API Agent* queries the Octopart distributor API to obtain structured part metadata and datasheet links.
   (ii) A *Datasheet Agent* performs a Google-based search to locate datasheet PDFs directly, prioritising high-confidence domains (e.g., manufacturer or distributor sites). And leverages Vision Language Models (VLMs) to extract the key electrical specifications from the retrieved datasheets.

3. A **Validation Module** enforces JSON-Schema constraints and physical range rules and flags low-confidence extractions for manual review.

4. Finally, a **Cost-Prediction Model** is trained on the extracted specifications, demonstrating the economic value of the pipeline's output.

1.5 Contribution

The thesis offers four concrete contributions:

1. A fault-tolerant multi-agent framework that integrates LLMs, web search, distributor APIs, and VLM parsing into a single orchestrated workflow.

2. Empirical benchmarks reporting MPN extraction accuracy, datasheet retrieval success, field-level precision/recall, and end-to-end latency.

3. A downstream cost-prediction experiment showing that structured specs extracted by the system improve price-estimation accuracy, thus underlining direct business impact.

1.6 Thesis Structure

The remainder of this document is organised as follows: Chapter 2 surveys literature on language-model agents, vision-language models, and document AI in procurement. Chapter 3 formalises the challenges inherent to QAF processing. Chapter 4 details the proposed multi-agent architecture, while Chapter 5 describes implementation specifics, including prompts, API wrappers, and validation logic. Chapter 6 presents the experimental setup and quantitative results. Chapter 7 demonstrates the downstream cost-modelling application. Chapter 8 reflects on limitations and industrial implications, and Chapterion 9 concludes the thesis with avenues for future work.

2 Related Work

This chapter surveys prior research relevant to the proposed multi-agent pipeline, grouped into four areas: (i) tool-using language-model agents, (ii) vision-language models for document understanding, (iii) multi-agent orchestration frameworks, and (iv) document-AI applications in procurement and engineering.

2.1 Tool-Using Language-Model Agents

Large language models have been extended with *tool use* capabilities to overcome context-length limits and provide up-to-date information. ReAct [Shu+23] interleaves chain-of-thought reasoning with executable actions, while the OpenAI function-calling interface popularised structured tool invocation in production systems. LangChain[1] offers an open-source abstraction layer for wrapping external tools (APIs, SQL, web search) behind LLM calls. AutoGPT and Voyager [Wan+23] push autonomy further but lack guarantees on reliability—highlighting the need for validation mechanisms such as those proposed in this thesis.

2.1.1 Origins of Tool Use in LLMs

2.1.2 ReAct-style Reasoning + Acting

2.1.3 Structured Function-Calling Interfaces

2.1.4 Autonomous & Memory-Augmented Agents

2.2 Vision–Language Models for Document Understanding

Document–analysis research has evolved through three overlapping stages: (1) *rule- and library-based PDF parsing*, (2) *layout-aware pre-trained transformers*, and (3) *large multimodal LLMs*. We review each strand and position our work in the third.

[1] `https://www.langchain.com`

2.2.1 Rule- and Library-Based PDF Parsing

Early pipelines extracted text and tables using Python libraries such as `PDFMiner`, `PyPDF2`, `pdfplumber`, and `Camelot/Tabula`.[2] Domain practitioners layered regular expressions, heuristic column detection, and X-Y–coordinate clustering on these low-level primitives. While effective for homogeneous reports or invoices, such systems break on

1. multi-column datasheets with mixed units,
2. vendor-specific typographic quirks (rotated headers, unusual glyphs),
3. text embedded inside nested tables or header/footer bands, and
4. heterogeneous page layouts within a single file.

They also provide no learned representation of visual context, forcing every format change to be patched manually.

2.2.2 Layout-Aware Pre-training

Transformer encoders that ingest token embeddings augmented with 2-D coordinates alleviate fragile heuristics. LayoutLMv3 [Xu+22] unifies text, layout, and image patches; StrucTexT [Li+21] adds contrastive pre-training for table structure. These models excel on benchmarks such as FUNSD and RVL-CDIP, yet published studies seldom target engineering datasheets whose key attributes are numeric ranges buried in dense specification tables.

2.2.3 Large Multimodal LLMs

General-purpose Llama3, Gemini, and GPT-4 Vision—combine a frozen vision encoder with a large language model, enabling open-vocabulary reasoning across arbitrary documents. Although early demonstrations highlight science articles and receipts, rigorous evaluation on *technical component datasheets* is scarce. Our work fills this gap: a GPT-4 Vision–based **Parsing Agent** is prompt-engineered to output *schema-validated JSON* containing technical parameters. This focus on reliability contrasts with prior VLM applications that prioritise free-form captioning or QA.

[2]See project pages: `https://github.com/pdfminer/pdfminer.six`, `https://github.com/py-pdf/pypdf`, etc.

2.3 Multi-Agent Orchestration Frameworks

MetaGPT [Hon+23] assigns specialised roles (product manager, engineer) to different LLM instances. CrewAI[3] and OpenAI's "Assistants" API provide abstractions for role-based coordination. Most studies focus on qualitative demos; quantitative benchmarks on noisy industrial tasks, such as QAF processing, are scarce. Our work contributes empirical results on orchestration efficacy (success) in a production-grade setting.

2.3.1 AutoGen

2.3.2 CrewAI

2.3.3 Langgraph

2.4 Research Gap

- VLMs demonstrate impressive captioning accuracy yet are rarely evaluated on fine-grained numeric specs inside technical PDFs.
- No prior work jointly optimises extraction *and* downstream business value (e.g. cost prediction) within an end-to-end industrial pipeline.

These gaps motivate the multi-agent system and evaluation strategy proposed in the following chapters.

[3] `https://github.com/joaomdmoura/crewAI`

3 Problem Setting

3.1 Characteristics of Quotation Analysis Forms (QAFs)

A Quotation Analysis Form (QAF) is a semi-structured document exchanged during early supplier negotiations. At BMW, QAFs appear in Excel sheets. Typical fields include:

- **Part Designation:** free-text description (e. g., "STM32F103C8 MCU 64 KB Flash TQFP48"),
- **Price and MOQ:** preliminary unit price, minimum order quantity,

The challenge lies in the Part Designation column, which often combines technical specifications and the manufacturer part number (MPN) into a single free-text field. This lack of structure, along with multilingual notations (e.g., Gehäuse", Temp. Bereich"), complicates rule-based parsing.

3.2 Core Challenges

C1: Inconsistent Identifier Extraction An MPN may be surrounded by package codes ("`-TR`"), voltage values, or marketing text, making pattern-matching unreliable. Example:

```
PI3USB9281ZLE_EX 20V Vbus Switch, TQFN-14, -40-85°C
```

C2: Missing Identifier-to-Parameter Mapping Even after correctly extracting an MPN, there is no comprehensive public or commercial database that reliably maps part numbers to their technical parameters—not even major distributors provide complete coverage. This forces systems to extract parameters directly from datasheets, introducing additional complexity and dependence on document parsing.

C3: Datasheet Discovery and Access Locating a reliable datasheet for a given MPN involves searching across:

1. manufacturer websites (high trust),
2. distributor platforms (moderate trust),
3. third-party mirrors (low trust, often behind captchas or broken links).

Selecting the most reliable source requires balancing trust, accessibility, and past success rates.

C4: Parameter Extraction from Unstructured Documents Datasheets are often long, unstructured PDFs with inconsistent formatting. Technical parameters may appear in dense tables, inline text, or charts, and standard extraction tools frequently miss context, units, or formatting—compromising accuracy.

C5: Parameter Alignment Across Sources Technical specifications for a given part may be scattered across multiple sources—for example, partially listed on a distributor's website and partially extracted from the manufacturer's datasheet. These representations often differ in terminology, format, precision, or units. Reconciling this fragmented information requires normalization, deduplication, and conflict resolution to generate a consistent and accurate parameter set.

C6: Provenance and Traceability Requirements For safety-critical or compliance-sensitive applications, it's not enough to extract technical parameters—each value must be traceable to its original source. This includes metadata such as the document name, page number, table or bounding box location, and source type (e.g., datasheet vs. distributor site). Lack of provenance undermines trust in the extracted data and complicates manual verification or auditing by engineers.

3.3 Formal Problem Definition

Let $q \in \mathcal{Q}$ denote a single QAF entry containing noisy text tokens. The task is to compute a validated specification record

$$\mathbf{s} = \big(\text{MPN}, \text{Voltage}_{\text{max}}, \text{Current}_{\text{max}}, \text{TempRange}, \ldots\big)$$

such that:

1. **Correctness:** each numeric field equals the value printed in the authoritative datasheet.
2. **Completeness:** mandatory fields are present; optional fields are `null` if not in the document.
3. **Provenance:** each field is traceable to its source, including its source, document name, page number.
4. **Latency:** average end-to-end processing time < 15 s per QAF.

3.4 Automation Requirements

Derived from the challenges and formal criteria, the target system must:

1. **R1: Noise-Tolerant Identifier Extraction** – Accurately extract MPNs from unstructured, noisy part designation strings, despite the presence of packaging codes, technical descriptors, or marketing language (addresses C1).
2. **R2: No-Schema Parameter Discovery** – Support parameter extraction even in the absence of a structured identifier-to-parameter mapping, by relying on downstream datasheet parsing (addresses C2).
3. **R3: Resilient Datasheet Retrieval** – Retrieve datasheets from a mix of trusted and unreliable sources, prioritizing success rate and content reliability while handling captchas, dead links, and inconsistent formats (addresses C3).
4. **R4: Multi-Source Parameter Fusion** – Normalize and reconcile parameter values found across datasheets and distributor sites, resolving differences in naming, units, and precision to generate a coherent specification set (addresses C5).
5. **R5: Structured Parameter Extraction** – Accurately extract parameter values from unstructured documents (PDFs), handling tables, prose, and visual elements using multi-modal models (addresses C4).
6. **R6: End-to-End Traceability** – Track and store detailed provenance for each parameter, including source URL, document page, and bounding box, to support validation, auditing, and engineer trust (addresses C6).

3.5 Mapping Challenges to Research Questions

Challenge	Requirement	Mapped RQ
C1	R1	RQ1
C4	R5	RQ2
C5	R4	RQ3
C6	R6	RQ4
(Cross-cutting)	R6	All RQs (system-level trade-offs)

This mapping clarifies how each experimental evaluation in later chapters directly addresses an identified pain point in BMW's procurement workflow.

4 System Architecture

This chapter describes the overall design of the proposed multi-agent pipeline, the responsibilities of each agent, their communication strategy, and the supporting infrastructure for storage, logging, and fault-recovery.

5 Implementation

6 Evaluation

7 Cost Model

8 Discussion

9 Conclusion

Abbreviations

List of Figures

List of Tables

Bibliography

[Hon+23] D. Hong, K. Zhang, H. Yang, Z. Li, X. Hu, W. X. Zhao, and J.-R. Wen. "MetaGPT: Meta Programming for Multi-Agent Collaborative Framework." In: *arXiv preprint arXiv:2308.00352* (2023).

[Li+21] X. Li, Y. Xu, L. Cui, S. Huang, F. Wei, and M. Zhou. "StrucTexT: Structured Text Understanding with Multi-modal Transformers." In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*. 2021, pp. 742–752. DOI: 10.1145/3447548.3467239.

[Shu+23] S. Shunyu Yao, J. Zhao, D. Yu, R. Anil, D. Yang, B. Y. Lin, Y. Zhao, W.-t. Yih, Y. Wang, and H. Hajishirzi. "ReAct: Synergizing Reasoning and Acting in Language Models." In: *arXiv preprint arXiv:2210.03629* (2023).

[Wan+23] T. Wang, Y. Zhang, J. Wang, Y. Zhu, T. Darrell, and D. Pathak. "Voyager: An Open-Ended Embodied Agent with Large Language Models." In: *arXiv preprint arXiv:2305.16291* (2023).

[Xu+22] Y. Xu, T. Huang, W. Lv, L. Cui, F. Wei, M. Zhou, and Y. Zhang. "LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking." In: *arXiv preprint arXiv:2204.08387* (2022).