AI-Driven Digitalization for an Automotive Component Manufacturer

Ruowen HE* Chunhui ZHENG* Jianbo HUANG*

Abstract

In the automotive industry, operational efficiency has been increasing because of digitalization and smart-factory transformation. To maintain competitiveness in this landscape, we developed an AI agent system that links retrieval-augmented generation to large language model. The system was applied to eight business scenarios, including knowledge retrieval and translation.

1. Introduction

The automotive industry is making rapid progress in improving operational efficiency through digitization and smart-factory transformation, and it is imperative for our company to apply these technologies in practice. In this study, we introduce our administrative initiatives at JATCO (Guangzhou) Automatic Transmission Ltd. In the area of administration, there are a wide variety of tasks such as knowledge retrieval, information organization, and contract review, and it is necessary to support not only specific tasks but also a wide range of them. In this study, we built a system that can be applied to actual operations by optimizing preprocessing, post-processing, and prompting, based on a method that combines retrieval-augmented generation (RAG) and a large language model (LLM).

1.1 Definition of terms

1) LLM [1]

Large language models are pretrained on large amounts of data and can perform a variety of language processing tasks (e.g., translation and analysis).

2) RAG [2]

A technology that retrieves relevant knowledge from external databases (e.g., terminology dictionaries and internal documents) in real time and augments the LLM to improve the accuracy of its responses.

3) Hybrid similarity

A retrieval evaluation index for a RAG system, which indicates the overall literal and semantic matching between the query and documents by calculating the weighted combination of keyword and vector similarities. Values range from 0% to 100%, with higher values indicating better overall matching.

4) Keyword similarity

A retrieval evaluation index for a RAG system that indicates the degree of lexical matching based on keyword overlap between the query and documents. Values range from 0% to 100%, with higher values indicating better lexical matching.

5) Vector similarity

A retrieval evaluation index for a RAG system that indicates the semantic similarity between a query and documents through high-dimensional vector transformation. Values range from 0% to 100%, with higher values indicating better semantic matching.

*Corporate Planning Department

2. Methodology and approach

2.1 Business analysis and the selection of core technology

In this study, we focused on high-frequency and high-volume work scenarios identified through an internal questionnaire. These include eight core business scenarios (Table 1), such as knowledge retrieval, translation, program coding, and contract review, covering a wide variety of practical cases, from general-purpose to specialized department-specific situations.

As a technical implementation method, RAG was employed as the core search engine responsible for document processing and vector [3] retrieval to provide high-quality knowledge retrieval. At the same time, an LLM was used as the basis for the language model service to provide the system's core functions of understanding language and generating texts.

This system presents and introduces novel processing methods for the pre- and post-processing for RAG and LLM to enhance the effectiveness of the final output in each scenario. As shown in Fig. 1, the overall structure of the system is the integrated flow of knowledge retrieval and language processing by linking RAG and LLM.

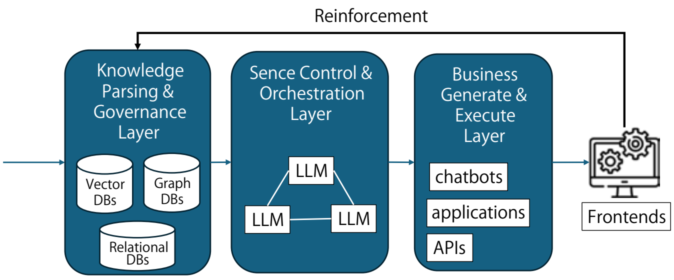


Fig. 1 Configuration of the system

2.2 Preprocessing

Preprocessing is a fundamental process that determines the accuracy, stability, and business compatibility of the RAG and LLM-based AI agent system. In the business scenarios targeted by this system, such as knowledge retrieval and translation, the documents to be processed are in a variety of formats, including PDF, Microsoft Word, and Microsoft Excel Note [Note1]. Hence, it is difficult to satisfy all business needs using a single preprocessing step.

In the preprocessing of this system, we used the "common infrastructure + scenario specialization" approach (Fig. 2) and designed preprocessing strategies according to the characteristics of each task. That is, common processes such as document structure analysis and semantic extraction were standardized, while specialized processes were developed by including the strategic conversion of the document format for knowledge retrieval, as well as structuring and reconfiguration of contents to preserve the structural elements of source documents and ensure semantic consistency.

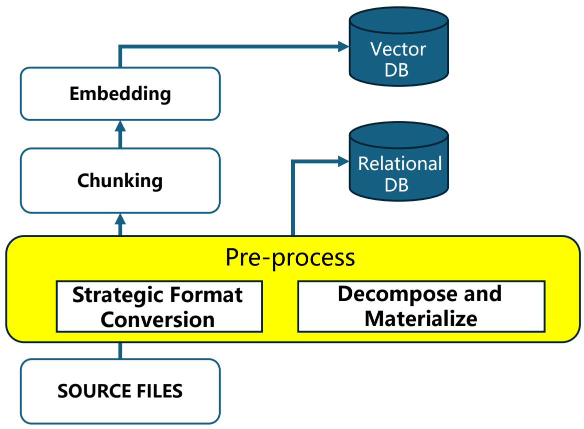


Fig. 2 Preprocess of RAG

Table 1 Matrix of departments and works

Department	Knowledge retrieval	Information organization	Program coding	Translation and Interpretation	Request and submission for approval	Contract review	Application proxy service	PC kitting
Production Engineering	R	R	R	I	I	I	I	-
Information System	A	A, R	A, R	A, R	R	R	R	A, R
Management	R	R	R	I	C	C	A	-
Finance	R	R	R	I	I	I	I	-
Legal Affairs	R	R	R	I	A	A	I	-
Purchasing	R	R	R	I	I	I	I	-
Manufacturing	R	R	R	I	I	I	I	-

RACI: I (Informed)
C (Consulted)
R (Responsible)
A (Accountable)

Note 1: Microsoft Word and Microsoft Excel are registered trademarks of Microsoft Corporation.

2.2.1 Strategic format conversion [8]

In a RAG system, the quality of document parsing has a direct impact on search effectiveness. Conventional methods directly parse documents in their original format (particularly Excel files), resulting in serious problems, such as "semantic fragmentation" and "loss of structural information." The reason is that the division of Excel files relies mainly on mechanical processing based on worksheets and a fixed number of rows, which hinders the understanding of matrix relationships among cells and the semantic linkage of data, resulting in the fragmentation of information and loss of structural linkage. Furthermore, Excel analysis often uses lightweight processing libraries and lacks the multistep processing mechanisms based on layout analysis and semantic understanding, which are widely used in PDF parsing. This makes it difficult to preserve the hierarchical structure of complex tables or logical relationships among data. To build a standardized processing infrastructure, the system first converts various documents into PDF files in a unified manner. PDF files have a fixed layout and stable structure, which are advantageous for preserving the original layout and semantic integrity of the document. Thus, the accuracy of the analysis is significantly improved over the direct processing of complex Excel source files.

2.2.2 Structure analysis and reconstruction of contents

Knowledge retrieval and translation are important application scenarios in the digitalization of business operations. Although both share a technical foundation in the layers of document parsing and semantic understanding, the complex tables, graphs, and other structural elements contained in Excel files are format-sensitive and prone to problems, such as broken layout, distorted meaning, and disconnected context, in a generic translation process.

There are two important data processing solutions to this problem.

1) Text elements (e.g., cells, comments, and graph labels) in Excel files do not exist in isolation, but their positions and combined states constitute an important semantic context. Therefore, Excel elements were extracted using an in-house Extractor and stored in a database (Table 2). Note that the structure shown here is a schematic and not the complete table structure of the actual business scenario.

Table 2 Structured tokens of the text

Idx	File name	Sheet name	Element type	Location	Row	Contents	Result
1	File 1	Sheet 1	Cell	A1	1	Test 1	
2	File 1	Sheet 1	Cell	A1	2	Test 2	
3	File 1	Sheet 2	Shape	Shape name	2	Test 1	

2) The Extractor has a function that allows customization of extraction patterns and filtering of characters according to the operational scenario. For example, when using LLMs, it is possible to exclude elements that do not need to be translated, such as simple "〇", "-", and numbers, to ensure translation accuracy and speed.

2.2.3 Prompt engineering

The source text is entered to be batch-translated and a prompt is created.

Example prompt:

1. Input format：{{Idx}和文原文};{{Idx}和文原文};...（Idxはインデクスの数字）
2. Output format：{{Idx} 中文翻訳 }};{{Idx} 中文翻訳 };...

Example of a text to be translated:

……

{{1} テスト 1};{{2} テスト 2};{{3} テスト 1};

……

Example of an AI response:

……

{{1} 测试 1};{{2} 测试 2};{{3} 测试 1};

……

Note that the prompts shown here are schematic examples based on Table 2, and not actual operational scenarios.

To input the translation from AI responses into the next process, the system first uses a regular expression (e.g., ‘\{\{(\d+)\}(.*?)\}’) [6] to match the fixed structure of "{{Idx}Translated text}". In the initial step, it detects structural breakdowns caused by the LLM's hallucination (e.g., `{{1}测 试1` (lack of closing brackets), `{{A}测试1}` (Idx is not a number), and `{{3}测试3}` (content consistency breakdown)).

Next, "verification of structural breakdowns" is performed based on the results of the regular expression analysis. Specifically, the number and type of detected breakdowns are determined, and if a threshold (e.g., one or more breakdowns) is exceeded, then the LLM is re-directed to "generate responses that comply with the structure" (re-execution design). This avoids the negative effects of structural breakdowns caused by the LLM's hallucination and guarantees the accuracy of the translation extraction.

To solve the problem of terminology mismatch, this study introduced a dynamic terminology reinforcement mechanism in the translation command generation phase (Fig. 3). The system searches the terminology dictionary in real time and embeds the identified terms and their in-house standard translations into the LLM prompts as mandatory constraints, thereby fixing the translation of keywords from the source side and ensuring terminology consistency throughout the whole text.

Optimally structured prompts are embedded in LLM requests to ensure stable outputs in cases such as large translation jobs.

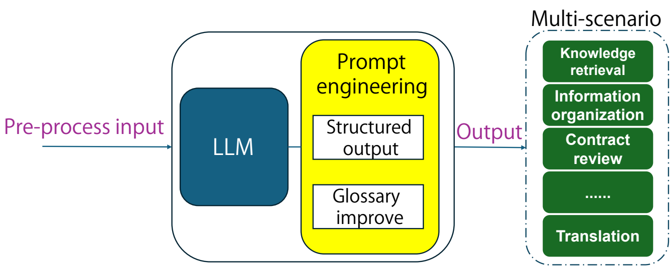


Fig. 3 Prompt engineering

2.3 Post-processing

2.3.1 Competitive optimal selection of dual LLMs [7]

To address the bias of a single model, this study employed parallel translation using dual LLMs and an automatic optimal-selection mechanism (Fig. 4). After completing the "phase matching" of the translation, the system performs comparison and scoring for each based on multidimensional criteria such as the degree of terminological matching, contextual accuracy, and naturalness of language flow, and automatically selects the best overall translation. This optimal selection mechanism effectively combines the strengths of different models to achieve significant improvement in translation quality.

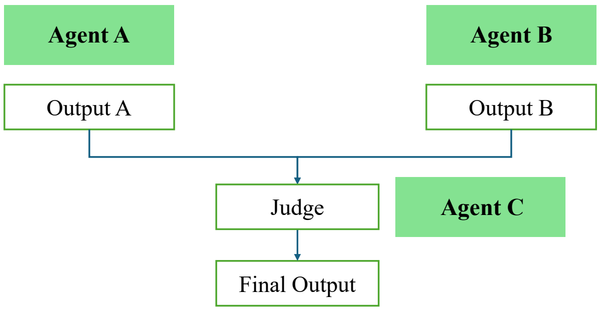


Fig. 4 Optimal selection of dual LLM outputs

2.3.2 High-precision image processing

This study also solved the problem of translating images in documents. Common image translation tools or application programming interfaces usually have difficulty preserving the visual layout of the original image. To solve this problem, a refined image processing subprocess was designed. The core steps and technical challenges are described below.

1) Optical character recognition [4], [5] and the structuring of text and coordinates

First, bulk extraction of embedded images in Excel was performed. Then the text content and its bounding box coordinates (A, B, C, and D) are recognized by optical character recognition (OCR). This process converts the image information into structured data containing text and coordinates, then establishes the basis for subsequent high-precision processing (Fig. 5).

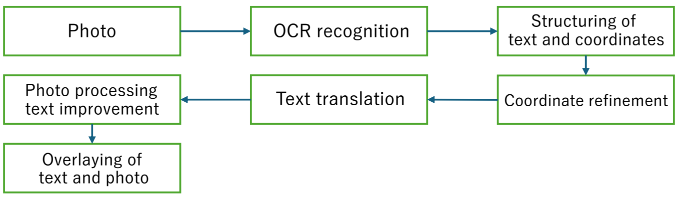


Fig. 5 Use of position information in text OCR

2) High-precision calibration of text coordinates

To address the problem of errors in the coordinate recognition by OCR, an automatic coordinate correction algorithm was designed. The coordinates were refined pixel-by-pixel using a "dual slider" mechanism. In addition, the final coordinates after refinement (A1, B1, C1, and D1) were determined by judging the validity of text recognition (e.g., correctness of character segmentation and reasonableness of blank areas) using the algorithm (Fig. 6).

3) Text translation, rewriting, and image composition

After obtaining the precise coordinates, the system performs LLM translation of the recognized text. Subsequently, based on the calibrated coordinates, the system renders the translated texts in the appropriate font and overlays them on the original image to complete the generation and reconstruction of a high-quality translated image.

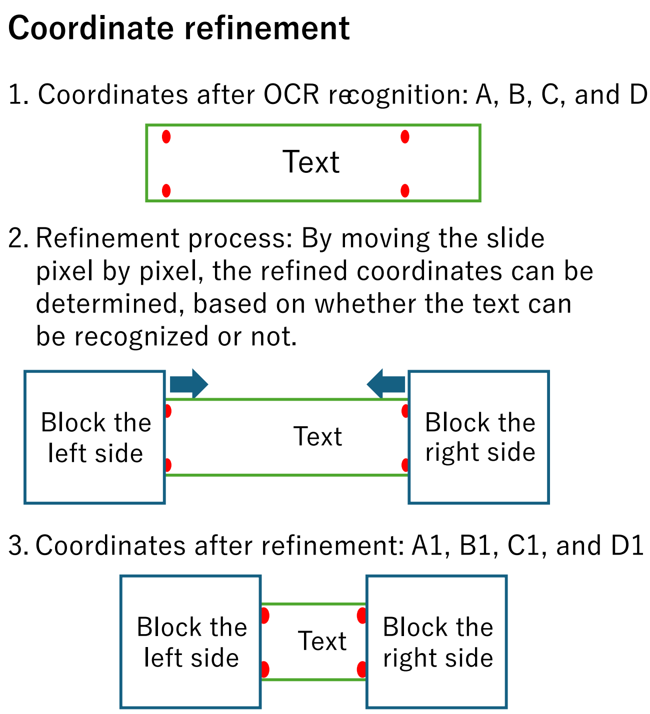


Fig. 6 Refinement of text position

3. Result

As discussed in Sections 2.2 and 2.3, the integration of preprocessing and post-processing with the AI agent system of this study (a combination of RAG and LLM) greatly improved the accuracy of the AI's responses.

3.1 Effects of preprocessing

To evaluate the effects of unifying documents formats into PDF in the process of normalizing specific file formats, a comparison experiment was conducted using vector searches (Table 3). The keyword similarity was consistent under the same query because of the same matching content of keywords in the content of the chunks. PDF chunks had higher vector similarity in most cases, demonstrating that their semantic structure was better preserved.

Table 3 Effect of file format conversion

File format	Average hybrid similarity [%]	Average keyword similarity [%]	Average vector similarity [%]
Excel	71.86	73.95	66.98
PDF	74.8	73.95	76.78

In this experiment, the performances of two representative LLMs (LLM A and LLM B) were compared in standard and terminology-enhanced prompts. The mean values of each group were calculated after multiple repetitions. As shown in Fig. 7, terminology injection improved the terminology usage rate from 32% to 98% in LLM A and from 22% to 86% in LLM B, resulting in a significant average improvement of 65 percentage points.

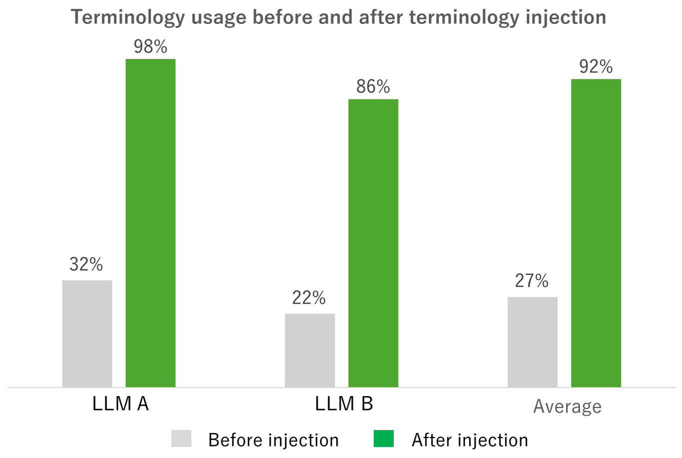


Fig. 7 Terminology injection effect

3.2 Effects of post-processing

We performed translation on a dataset using two different LLMs and found differences between the outputs. The average translation accuracy of a single LLM was 96%. However, by introducing an optimal selection mechanism using a third LLM to determine the translation differences between the two LLMs, the overall translation accuracy increased to 97%. This result suggests that it is possible to improve translation accuracy through a cooperative mechanism using multiple LLMs.

For high-precision image processing, the coordinate refining process significantly improved the consistency of the translation from Chinese to Japanese, as shown in Fig. 8.

Fig. 8 Coordinate precision effect

Finally, through the translation of a large volume of technical documents used in this development, an independent accuracy-oriented technical route was designed, and a mixed strategy of "parallel processing with dual LLMs, semantic enhancement (RAG), structured control, and optimal selection of results" was adopted. The overall test results showed that the consistency in the translation of internal terminology and overall quality of the translation were significantly improved, and the structural elements were effectively preserved. Compared with the previous business process, the total number of labor hours required for translation was also significantly reduced. In particular, the translation of texts within images has effectively eliminated traditional issues such as coordinate misalignment, enabling the stable provision of high-quality translation necessary for accurate communication and sharing of technical information.

By applying the above methods, the system achieved both "accuracy improvement" and "efficiency improvement" in a wide range of administrative tasks at an automotive parts manufacturer.

4. Discussion

In this initiative for improved system efficiency, the results were manifested by the cooperative mechanism between RAG and LLM. Specifically, RAG searched and reinforced terminology and in-house knowledge from external databases in real time, which suppressed the LLMs' hallucination and improved the consistency of terminology in translation and the accuracy of contract reviews. At the same time, in knowledge retrieval operations, high-quality knowledge recall by RAG resulted in significant time savings compared with conventional manual retrieval. Thus, knowledge-retrieval capability is the foundation for both business compatibility and processing accuracy of the AI system; it is an essential element for the practical application of AI collaboration and its utilization.

Accurate semantic chunking is a core factor in improving the retrieval accuracy of RAG. Structured data-preprocessing approaches-such as extracting content contained within shape frames or image elements during Excel translation-offer new ideas for semantic chunking in other file formats.

5. Conclusion

The results of this study demonstrated that the cooperative model of knowledge retrieval capability by RAG and language processing capability by LLM is highly adaptable and effective in improving the efficiency of a wide variety of administrative tasks in the automotive parts manufacturing industry. This model provided a solution for both automation and quality assurance of complex document processing, providing a new perspective on the practical application of AI technology in the smart-factory transformation of the automotive industry.

6. Summary

In this study, an AI agent system that links RAG and LLM was used to digitalize tasks and reduce labor hours in administrative areas. The use of AI technology in our company is not limited to the administrative department, but is also applied to the production floor, where image processing and deep learning are used to automate inspections of machining processes.

After one year of operation of the AI agent system, it was found that the total labor hours saved in eight work scenarios reached 1,837 hours per year, including translation and interpretation (722 h) and knowledge retrieval (381 h), as shown in Table 4.

Table 4 Labor hour reduction effect

Core business scenarios	Reduction in man-hours (h)
Translation and interpretation	722
Knowledge retrieval	381
Application proxy service/PC kitting	243
Contract review/request and submission for approval	240
Program coding	231
Information organization	20

The following four challenges need to be addressed in the future.

1) Further application of the pre- and post-processing architecture of this study should be considered for wider business scenarios.
2) Output stability and processing speed of each module should be improved further.
3) The architecture and implementation strategy should be adjusted from time to time in keeping with the development of LLM technology.
4) Aimed at advanced applications, role-type agents should be developed, such as assistants for office work execution, meeting moderators, educational lecturers, predictive maintenance assistants for facilities, and assistants for designing parts.

7. References

[1] Alammar, J. & Grootendorst, M. (2024). Hands-On Large Language Models. O'Reilly Media. (Official code repository: https://gitcode.com/GitHub_Trending/ha/Hands-On-Large-Language-Models).

[2] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33 (NeurIPS 2020). arXiv Preprint arXiv:2005.11401. https://doi.org/10.48550/arXiv.2005.11401.

[3] Johnson, J., Douze, M., & Jégou, H. (2019). FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors. Research Facebook, arXiv Preprint arXiv:1702.08734. https://doi.org/10.48550/arXiv.1702.08734.

[4] Cheng Cui, Ting Sun, Manhui Lin, et al. PaddleOCR 3.0 Technical Report [Online]. 2025. arXiv:2507.05595 [cs.CV]. https://arxiv.org/abs/2507.05595.

[5] Cheng Cui, Ting Sun, Suyin Liang, et al. PaddleOCR-VL: Boosting Multilingual Document Parsing via a 0.9B Ultra-Compact Vision-Language Model [Online]. 2025. arXiv:2510.14528 [cs.CV]. https://arxiv.org/abs/2510.14528.

[6] Jeffrey E. F. Friedl: Mastering Regular Expressions, 3rd Ed., Sebastopol, CA, O'Reilly Media, Inc., 2006, 515 pages. ISBN: 0-596-52812-4

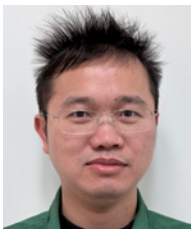
[7] Zhang, Y., Li, J., & Chen, W. (2023). JudgeLLM: Evaluating and Selecting LLM Outputs via a Specialized Judge Model. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8924–8938, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.521

[8] Zixuan Ke, Yucheng Wang, Yichao Zhou, Jia Li, Yutao Zhu, Yujiu Yang, Yi Liu, Ninghao Liu, Haiyun Jiang, Wayne Xin Zhao, & Zhicheng Dou. (2023). A Survey on Table Understanding in Natural Language Processing. arXiv Preprint arXiv:2305.13095. https://doi.org/10.48550/arXiv.2305.13095

Authors

Ruowen He

Chunhui Zheng



Jianbo Huang