Modification of the Test Protocols Generation with the Knowledge Graphs*

Vitalii Lavrenko[1,†], Olena Shytikova[1,*,†] and Galyna Tabunshchyk[2,*,†]

[1] National University Zaporizhzhia Polytechnic, Universitetska Str. 64, 69063, Zaporizhzhia, Ukraine
[2] Ruhr-Universität Bochum, Universitätsstr. 150, 44780 Bochum , Germany

Abstract

Using Large Language Models (LLMs) to automate routine tasks in processing structured and unstructured data – specifically, test protocols for stationary gas turbine units (GTUs) – shows great promise. However, processing tabular data remains challenging. To improve search efficiency and logical inference, the authors propose integrating knowledge graphs with prompt templates. This approach allows for the representation of table structures of any complexity, and a modified framework based on this integration has been developed. This framework outlines the workflow for handling GTU test protocol input and extracting knowledge to generate output documents. To improve output efficiency, specific prompt templates are proposed for data extraction based on document structures defined within a knowledge graph. Experiments demonstrate that performance improved by 15% compared to the basic model.

Keywords

operating parameters, consolidated test protocol, summary data table, RAG, knowledge graph, prompt

1. Introduction

Stationary gas turbine units (GTUs) are complex technical systems where automated control systems (ACS) monitor numerous parameters for control algorithms and to indicate the health and safe operation of the plant [1].

During GTU testing, the operability of the entire plant and all its components is verified, including parts, subassemblies, units, and functional systems. Tests are carried out across all operating modes specified by technical regulations. Critical parameters of GTU operation are simultaneously recorded because they are characteristic of each operating mode.

One of the primary functions of ACS is to record all GTU operational parameters in archive files for subsequent retrieval and analysis. However, current practice still involves manually recording key plant parameters in test protocols, including descriptions of adjustments made, external factors that may influence results, and identification of responsible personnel. These parameters are documented at every mode change or anomalous situation. All operating parameters are compared against permissible value ranges to confirm proper functionality.

Upon completion of the active testing phase, the data processing and result analysis stage begins [2]. Numerous test protocols are processed to create a consolidated protocol, where data is aggregated over the entire active phase. The consolidated test protocol then serves as the primary source for a summary data table, which is used to populate technical forms and documentation for the GTUs. Figure 1 illustrates the conventional data flow between these documents.

* CMIS-2026: The Ninth International Workshop on Computer Modeling and Intelligent Systems, 5 of May 2026, Zaporizhzhia, Ukraine

[1*] Corresponding author.
[†] These authors contributed equally.
zckernel@gmail.com (V. Lavrenko); helenshitikova@gmail.com (O. Shytikova); galina.tabunshchik@gmail.com (G. Tabunshchyk)
0009-0003-0489-9156 (V. Lavrenko); 0000-0003-4881-4882 (O. Shytikova); 0000-0003-1429-5180 (G. Tabunshchyk)

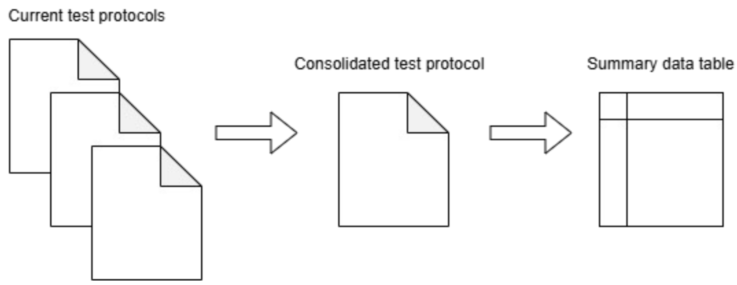

Figure 1: Data flow between documents.

The substantial volume of routine manual labor described above presents a significant optimization opportunity through automation. This research addresses the challenges and problems the authors encountered while automating data processing from GTU test protocols.

Section 2 provides an analysis of modern scientific works devoted to large language models (LLMs), with a focus on the retrieval-augmented generation (RAG) approach for analyzing technical data from gas turbine units.

Section 3 presents tasks developed specifically to increase the accuracy of extracting data from protocols containing complex tables. Using knowledge graphs that can represent table structures of any complexity is proposed. The work's overall goal is also formulated here.

Section 4 describes a modified framework based on a knowledge graph. This section outlines the process of handling input GTU test protocols and extracting knowledge to generate output documents. Section 5 provides information about experiments conducted using this framework to investigate data extraction through the integration of knowledge graphs, prompt templates, and LLMs. Finally, Section 6 summarizes the research findings.

2. State of Art

Large language models (LLMs) have been widely implemented across various fields and have demonstrated significant results (3–4). These systems can process textual data, interpret content, extract key information, and generate human-like responses. They also perform effectively in Natural Language Processing tasks. To enhance reliability and minimize errors, the Retrieval-Augmented Generation (RAG) approach is used. This approach generates answers based on previously retrieved relevant sources [5]. One study analyzed how this approach affects LLM performance quality and investigated the role of query precision in shaping results.

However, LLMs have several disadvantages, such as requiring substantial computational power and raising concerns about data security. Researchers addressed the use of local open-source LLMs, which support local hosting and ensure data confidentiality. This is critical for the present study because the GTU test results are proprietary and not publicly accessible, especially in the context of experimental development and high market competition.

However, LLMs have several disadvantages, including their substantial computational power requirements and data security concerns. Researchers addressed the use of local, open-source LLMs, which support local hosting and ensure data confidentiality. This is critical for the present study because the results of GTU tests are proprietary and not publicly accessible, particularly in the context of experimental development and intense market competition.

For GTUs, model training may be hindered by the limited number of available test protocols, which is common for specialized, low-volume industrial products. Nevertheless, research [8] demonstrates the effectiveness of LLMs with small data samples, showing that limited datasets can provide an acceptable level of quality.

Another essential aspect is the structural analysis of documents, which enables the precise identification of text fragments and tables while preserving the logical consistency of the content. Works [9, 10, 11] propose solutions to increase the accuracy of these tasks. Poor formatting of the input text can also lead to a partial loss of context and erroneous results. This is due to the structural gap between tabular data and natural language, which hinders the application of natural language reasoning. Converting table data into a linear, sequential text format (serialization) is essential for LLM performance and bridges this gap. Serialization typically comprises two main phases: partitioning and parsing, followed by data retrieval and extraction [12].

Classifying tables by their structure, internal relationships, and orientation enables the conversion of raw tabular data into structured knowledge graphs (see [13]). Work [14] presents a method for the automated creation of structured documents using a combination of LLMs and knowledge graphs. The authors address the issue of low AI accuracy in the industrial sector by implementing a step-by-step generation system based on logical reasoning chains and data extraction from professional knowledge bases. This approach is highly relevant for stationary gas turbine units.

One of the solutions which allows to improve the content accuracy generated by LLM is knowledge graph. The knowledge graph (KG) is utilized for its ability to store and retrieve information hierarchically, which is more effective than simple keyword matching [14]. In [9] authours use KG to preserve structured context, which is often lost in traditional vector-based retrieval systems. Authors showed that by integrating RAG with KG, systems can perform more complex reasoning and reduce the "hallucinations". KG are also used to model the internal structure of tables [15]. But the authors in [9] also come to the conclusions, that LLMs often struggle to determine the contextual interrelationship between text and tabular information provided in a graph or retrieval context, which can lead to a failure in retrieving the correct data segments.

3. Problem statement

In study [16], the authors considered implementing Retrieval Augmented Generation (RAG) to process gas turbine unit test results, which consist of mostly complete structured tables.

Initial experiments in processing GTUs test protocols using Large Language Models (LLMs) via the RAG method demonstrated low performance.

Among the identified challenges were the processing of handwritten Cyrillic text and the inherent complexity of extracting data from protocols stored in tabular formats. Consequently, the primary focus was directed toward increasing the accuracy of data retrieval from tables. This can be achieved by representing table structures through knowledge graphs and organizing structured template-based queries to the LLM.

An analysis of the protocol structures revealed a clear hierarchical organization, where information is ordered from the general to the specific. Such a well-defined structure, characterized by the absence of chaotic or cross-links between subordinate levels, aligns effectively with a hierarchical knowledge graph. Figure 2 illustrates the knowledge graph representing the hierarchical structure of a test protocol.

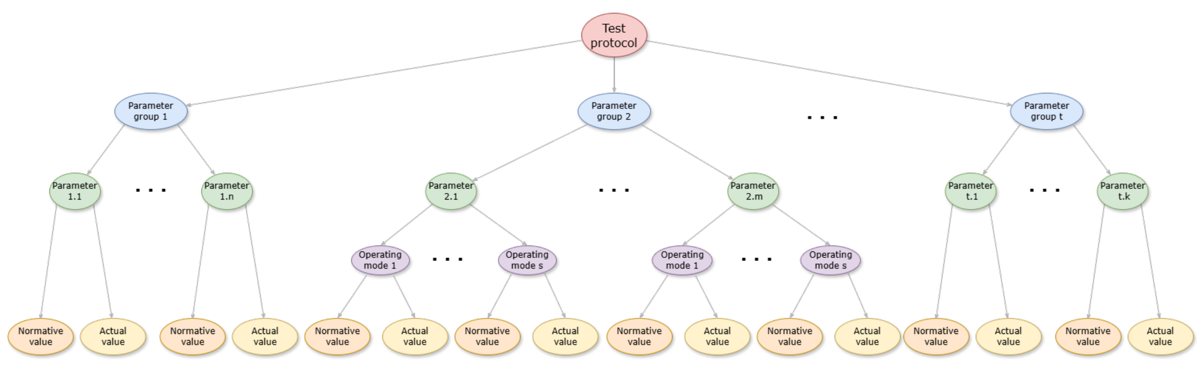

Figure 2: Knowledge graph of a test protocol.

Due to its hierarchical structure (protocol – parameter group – operating mode – value), the graph enables an accurate search for relevant information based on similarity. This ensures the traceability and interpretability of the results produced by the artificial intelligence.

On the other hand, the knowledge graph can enhance the efficiency of constructing prompt templates, transforming them from fragmented text queries into dynamic, context-sensitive structures. The knowledge graph allows for decomposing a complex prompt into sequential steps that correspond to the data hierarchy (e.g., parameter group – parameter – operating mode – actual value). This approach ensures a logical generation sequence. Furthermore, the prompt template is populated with standardized names and formulations through the unification and standardization of terminology.

The goal of this research is to increase the efficiency of data retrieval and logical inference based on test protocols stored in tabular form by integrating knowledge graphs and prompt templates with large language models.

4. Methodology

Based on the developed knowledge graph of the test protocol (Figure 2), a structure for the GTUs consolidated test protocol is proposed, as illustrated in Figure 3.

Furthermore, a modified framework based on the knowledge graph has been developed (Figure 4). The defined structure of the consolidated test protocol is utilized to mark up the input protocol, enabling the identification of structural features within the semi-structured document. This process includes the definition of classes (parameter groups), entities (the operating parameters), relationships between them, and properties (operating modes and corresponding parameter values).

Subsequently, knowledge extraction is performed: entities and properties are extracted through direct definition, while specific entities are further refined through tokenization. Finally, knowledge fusion is executed by integrating these entities to construct a comprehensive knowledge graph of the GTUs test protocols. Due to the tabular format of the test protocols, the majority of entities, attributes, and relationships can be identified and extracted based on the structural features of the table (Figure 3).

The resulting knowledge graph is then used to generate the output document – summary data table of test – which is also constructed based on the structural description (Figure 5). The same prompts working for any input table.

General view of the prompt (colors according to Figure 2):

Extract within parameter group – XXXXXXXXX – for parameters – XXXXXXX – on operating modes – XXXXXXX (if necessary) – only corresponding actual values

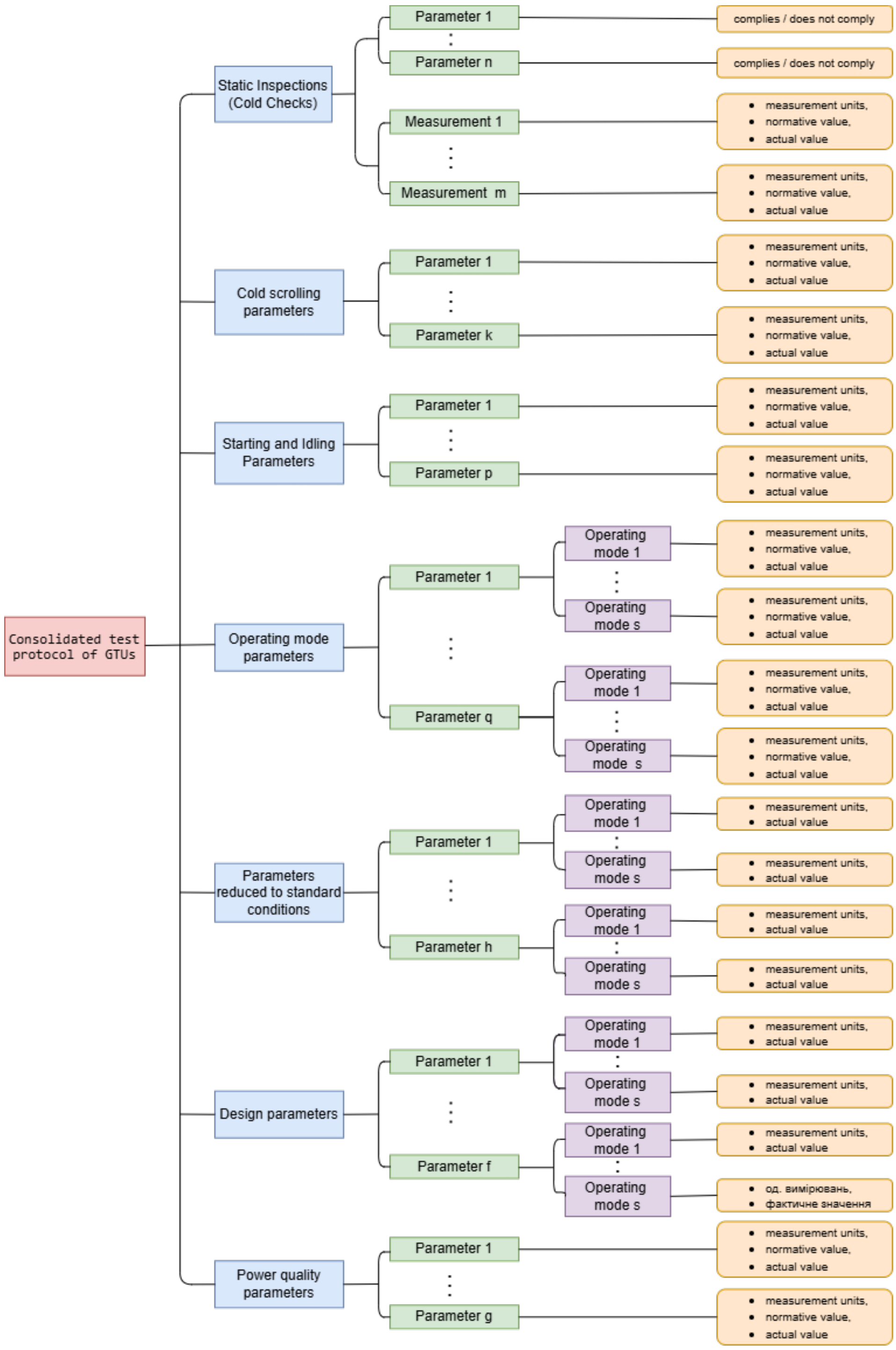

Figure 3: Structure of the GTUs consolidated test protocol.

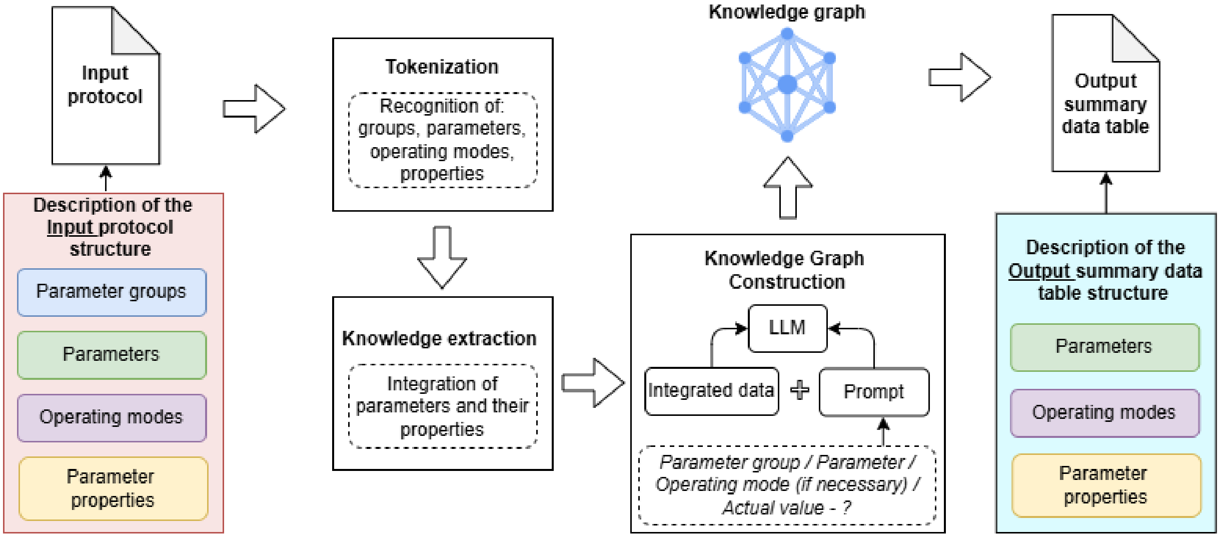

Figure 4: Modified framework based on the knowledge graph.

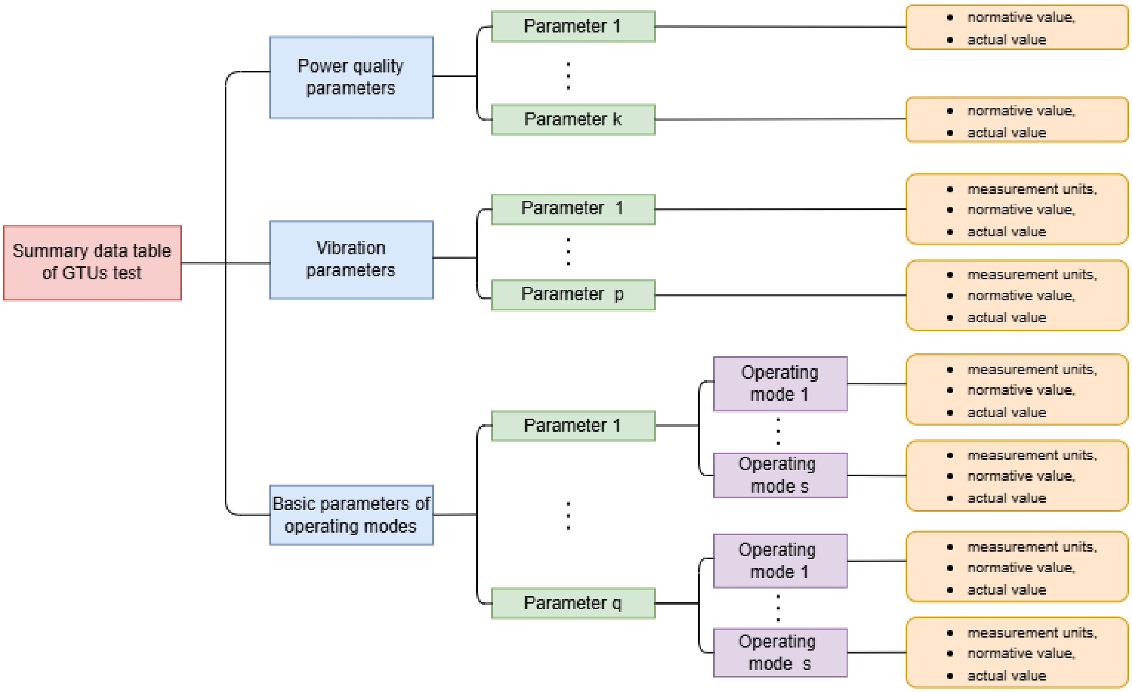

Figure 5: Structure of the summary data table of GTUs test.

5. Experiments

The implementation includes a retrieval-augmented mapping of converted CSV protocols with a knowledge graph featuring hybrid search and reranking. Since building domain-specific graphs with LLMs is difficult because traditional lexicon-based tokenization tools perform poorly on professional industrial vocabulary, the knowledge graph was created manually for the consolidated protocol. It was stored in observation-centric JSON with strict validation and full provenance, enabling reproducible extraction and analysis across evolving CSV protocols.

For each retrievable node (column, subcolumn, upper node, parameter, operating mode), we generate a compact textual 'node card' containing the ID, type, label and a one-sentence role/definition. Multilingual sentence embeddings suitable for Cyrillic text were used (IntFloat/Multilingual-E5-Base (E5)) [15, 17]. Retrieval-tuned embedding models were trained with instruction prefixes that inform the encoder that the text is a document. Using the same prefixes at inference time keeps the embedding spaces aligned and improves retrieval accuracy. The embeddings are L2-normalised, and cosine similarity is implemented as an inner product in a Facebook AI Similarity Search (FAISS) [18]. For each query token (e.g. the header 'Actual'), a BM25 search is run to find the top-k candidates with an exact or near-exact text match in order to create the candidate pool. Next, reranking is performed.

The Ollama Server is used for the following steps [19]: first, to parcel prompts; then, to select the node that best matches the knowledge graph. The hybrid retriever then narrows down the candidate set for groups, parameters, and modes. This makes the Ollama choice more precise. Values are fetched deterministically from the knowledge graph.

The examples of the prompts were following.

Prompt for Edge 1 Figure 5

1. Extract within parameter group – Power quality parameters – for all parameters – only corresponding actual values

Prompt for Edge 2 Figure 5

2. Extract within parameter group – Basic parameters of operating modes – for parameters – Vibration – on operating modes – Operating mode$_1$, ... , Operating mode$_s$ – only corresponding actual values

Prompt for Edge 3 Figure 5

3. Extract within parameter group – Basic parameters of operating modes – for parameters – Fuel consumption * – on operating modes Operating mode$_1$, ... , Operating mode$_s$ – only corresponding actual values

* – Extract prompt with the following parameters

Power turbine speed
Gas temperature downstream of the turbine
Oil pressure at the gas turbine inlet
Oil pressure at the gearbox inlet
Oil pressure at the turbogenerator inlet
Oil pressure at the Torquemeter system inlet
Voltage
Electric current
Current frequency

General improvements to the basic model were achieved at a rate of 15%. One of the most common problems is the program returning empty cells instead of values, which will be addressed in future work.

6. Conclusions

The implementation of RAG technology for generating reporting documentation from GTUs test protocols enables the effective automation of industrial processes while offering users an intuitive interface. The importance of local hosting is emphasized to ensure data confidentiality when processing test protocols.

At the same time, processing tabular structures within this task remains a significant challenge, necessitating the application of specialized preprocessing methods for input documents to ensure the validity of the results. The authors propose the integration of a knowledge graph and structured prompt templates with large language models to enhance the efficiency of search operations and logical data extraction.

A modified framework based on a knowledge graph has been developed, which leverages detailed descriptions of the tabular structures found in both input and output documents. Prompt templates for data extraction are proposed, which are based on the developed graphs of the structure of input and output documents.

Experimental research proves the positive impact of integrating knowledge graphs and prompt templates with large language models compared to previous experiments. Search accuracy was increased by 15%.

Future work will focus on implementing the full report generation sequence and development of the graph database.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini in order to: Grammar and spelling check. After using these tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Y. Shitikova, G. Tabunshchik, GTU Tests Results Monitoring System, in: Proceedings of the XIth International Conference, TCSET'2012, Publishing House of Lviv Polytechnic, Lviv, 2012, p.457.

[2] O. Shytikova, G. Tabunshchyk, Automated decision support system for GTU tests process, Central European Researchers Journal Vol. 1, Issue 2 (2015) 51-56.

[3] M. Raza, Z. Jahangir, M.B. Riaz, M.J. Saeed, M.A Sattar, Industrial applications of large language models, Scientific Reports 15:13755 (2025) 1–23. doi: 10.1038/s41598-025-98483-1.

[4] Z. Chkirbene, R. Hamila, A. Gouissem, U. Devrim, Large Language Models (LLM) in Industry: A Survey of Applications, Challenges, and Trends, in: Proceedings of the 21st International Conference on Smart Communities: Improving Quality of Life using AI, Ro-botics and IoT, HONET'24, IEEE, 2024, pp. 229–234. doi: 10.1109/HONET63146.2024.10822885.

[5] A. Mokash, B. Puthuparambil, C. Daniel, T. Hanne, Analysis of Large Language Models for Company Annual Reports Based on Retrieval-Augmented Generation, Information 16(786) (2025) 1–15. doi: 10.3390/info16090786.

[6] S. Xiong, C. Ouyang, Research on Document Layout Detection and Description Method for Retrieval-Augmented Generation of Large Language Model, in: Proceedings of 9th International Conference on Robotics, Control and Automation, ICRCA'24, IEEE, 2024, pp. 367 –370.
doi: 10.1109/ICRCA64997.2025.11011072.

[7] Khant Ko, Thwet Yin Nyein, Khine Khine Oo, Thant Zin Oo, Thet Thet Zin, Retrieval Augmented Generation for Document Query Automation using Open source LLMs. in: Proceedings of 5th International Conference on Advanced Information Technologies, ICAIT'24, IEEE, 2024. doi: 10.1109/ICAIT65209.2024.10754919.

[8] M. Sarioglu1, G. Sariyer, M.E. Sozen, LLM-based embeddings for clustering and predicting integrated reporting quality levels of companies, Discover Computing 28:95 (2025). doi: 10.1007/s10791-025-09590-6.

[9] M. Song, Enhancing RAG performance by representing hierarchical nodes in headers for tabular data, IEEE Access 13 (2025) 85072–85083. doi: 10.1109/access.2025.3569872.

[10] F. Dhanani, M. Rafi, Table Talk, in: Proceedings of 26th International Multitopic Conference, INMIC'24, IEEE, 2024, pp. 1–6. doi: 10.1109/inmic64792.2024.11004342.

[11] S. Raja, A. Mondal, CV Jawahar, Visual Understanding of Complex Table Structures from Document Images, arXiv, 2021. doi: 10.48550/arXiv.2111.07129

[12] Y. Sui, M. Zhou, M. Zhou, S. Han, D. Zhang, Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM'24, ACM Press, Mérida, Yucatán, Mexico, 2024. doi: 10.1145/3616855.3635752.
[13] J. Liu, Y. Chabot, R. Troncy, VP. Huynh, T. Labbé, P. Monnin, From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods, Web Semantics: Science, Services and Agents on the World Wide Web 76 (2023) 100761, Elsevier B.V. doi: 10.1016/j.websem.2022.1007611570-8268/.
[14] F. Shi, L. Chen, M. Zhou, Y. Zhao, Y. Zheng, A stepwise intelligence generative method for structured maintenance guidance documents based on knowledge graph augmented LLM, Advanced Engineering Informatics 67 (2025) 103523, ISSN 1474-0346. doi: 10.1016/j.aei.2025.103523.
[15] G. Yang, Y. Yan, D. Zeng, Y. Li, J. Zhang and W. Cao, Retrieval Augmented Generation for Tables in Textual Documents based on Graph Representation Learning, in: Proceedings of the International Conference on Industrial Technology, (ICIT'24), IEEE, Wuhan, China, 2025, pp. 1-6. doi: 10.1109/ICIT63637.2025.10965127.
[16] O. Shytikova. N. Oblisz, G. Tabunshchyk, A. Fay, Application of LLM for the Generation of the Testing Reports, in: Proceedings of 23rd International Conference on Smart Technologies & Education, STE'26, Transilvania University of Brașov, Romania, 2026, pp. 1–8.
[17] D. Li, K. Bi, J. Guo, X. Cheng. Tailoring Table Retrieval from a Field-aware Hybrid Matching Perspective, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2025, pp. 27693–27704.
[18] Welcome to Faiss Documentation – Faiss documentation. URL: https://faiss.ai/ .
[19] Ollama. (o. D.). GitHub - ollama/ollama: Get up and running with OpenAI gpt-oss, DeepSeek-R1, Gemma 3 and other models. GitHub. URL: https://github.com/ollama/ollama.