Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

AutoJSA: A Knowledge-Enhanced Large Language Model Framework for Improving Job Safety Analysis

Shuo Xu[a], Jinsong Zhao[a]*

[a] Tsinghua University, Department of Chemical Engineering, Beijing, China
* Corresponding Author: jinsongzhao@tsinghua.edu.cn.

ABSTRACT

Job Safety Analysis (JSA) is critical for proactively identifying workplace hazards, assessing their potential consequences, and implementing effective control measures. However, traditional JSA methods can be inefficient and prone to errors, particularly in complex industrial environments. This paper introduces AutoJSA, a knowledge-enhanced framework that leverages large language models (LLMs) to automate and optimize the JSA process. We collected 73 high-quality JSA reports from a chemical engineering company and divided the JSA workflow into three key tasks: hazard identification, consequence identification, and control measure generation. Two approaches — fine-tuning and retrieval-augmented generation (RAG) — were employed on a base LLM (GLM-4-9B-Chat) to adapt it for these domain-specific tasks. Experimental results demonstrate that both fine-tuning and RAG significantly improve task performance relative to the unmodified model, with fine-tuning generally providing larger gains. While RAG offers the advantages of dynamic knowledge retrieval and simpler implementation, it yields more modest improvements compared to fine-tuning. Moreover, a direct combination of fine-tuning and RAG did not show additional benefits under the current approach. Overall, AutoJSA effectively addresses the limitations of traditional JSA by increasing the accuracy and efficiency of safety analysis, laying the groundwork for fully automated JSA report generation. The findings underscore the potential of advanced artificial intelligence and natural language processing techniques to enhance workplace safety management in complex and rapidly evolving industrial settings.

Keywords: Artificial Intelligence, Job Safety Analysis, Large Language Model

1. INTRODUCTION

In the chemical process industry, tasks such as calibrating temperature control systems, replacing seals on reactors, and conducting maintenance on pumps and compressors are essential for maintaining safe and efficient operations. However, these activities can expose workers to hazards like electrical shock, high-temperature burns, or poisoning. Job Safety Analysis (JSA) is crucial for mitigating these risks by systematically examining each task, identifying potential hazards, and implementing effective control measures [1].

Heinrich is considered the first scholar to introduce the term Job Safety Analysis [2]. Since then, JSA has been widely applied in industries such as process engineering, construction, and healthcare [3], with various guidelines available for conducting JSA [4,5]. A comprehensive JSA typically includes four key components: Job Description, Hazard Identification, Consequence Identification, and Control Measures Generation. JSA improves workplace safety by identifying and addressing potential risks before work begins, helping to reduce accidents and injuries [6,7], and identifying potential human errors to ensure quality performance [8].

However, traditional JSA faces several challenges: it may fail to identify all potential risks, especially those arising from unforeseen circumstances or complex scenarios [9]. Additionally, it is time-consuming, particularly in dynamic environments [10], and there can be significant variations in hazard identification quality, as different individuals may have different expertise levels [11].

To address these challenges, researchers have explored AI methods to automate JSA. For example, one study integrated JSA with a modified Petri net for non-

routine operations, which helped identify sequential anomalies. However, its static and semi-quantitative nature limits dynamic risk simulation and the ability to account for complex interactions [12]. Another study developed a knowledge graph for automatic JSA using GRAKN.AI, which provided solid risk assessments, but it relied on pre-existing JSA documents and lacked real-time data or natural language support [13]. Similarly, a third proposed an ontology-based framework combined with Building Information Modeling for automated JSA. However, it overlooks dynamic factors, such as site layout changes, and requires manual rule adjustments [14]. While these approaches are sophisticated, they often suffer from limited accuracy and adaptability due to their reliance on expert rules, structured data, and static models, which can miss subtle or emerging risks and restrict cross-domain applicability.

Large language models (LLMs), like ChatGPT and GPT-4 [15], present a solution by handling unstructured text, detecting new risks, and adapting across domains—making them ideal for automating hazard identification and safety reporting. Although some studies have explored the application of LLMs for similar tasks [16-18], systematic evaluation of LLMs for the entire JSA process remains limited.

This paper introduces AutoJSA, a knowledge-enhanced LLM framework using fine-tuning and retrieval-augmented generation (RAG) techniques to improve JSA accuracy, efficiency, and adaptability in complex industrial environments.

2. METHODOLOGY

2.1 Framework Overview of AutoJSA

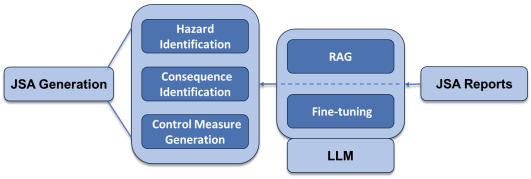

Figure 1. Framework overview of AutoJSA.

Figure 1 illustrates the framework of this study, which aims to generate JSA reports using LLMs. The dataset consists of high-quality JSA reports for training, validation, and testing.

The JSA report generation is divided into three subtasks: hazard identification, consequence identification, and control measure generation. Each task has corresponding JSA reports split into training, validation, and test sets.

Fine-tuning and RAG are applied to each sub-task. Fine-tuning optimizes the base model for task-specific data, while RAG enhances information retrieval and output generation using a structured knowledge base. Both methods are evaluated through a predefined framework to assess performance under different parameters.

The best-performing parameters, selected from validation results, are tested on the test set to compare the fine-tuned model, RAG-enhanced model, and base model. The evaluation will determine the effectiveness of fine-tuning and RAG in improving JSA report generation.

2.2 Data Description

The data used in this study consists of 73 high-quality JSA reports obtained from a chemical engineering company. These reports cover various jobs and provide a detailed safety analysis for each step of the job. Each JSA report is composed of two main sections: JobInfo and JSA. A detailed description of the data structure can be found in **Figure 2**.

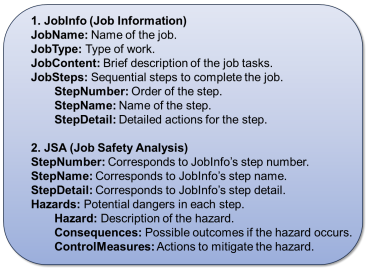

Figure 2. Detailed description of JSA report.

These reports are divided into three parts: 58 for training, 8 for validation, and 7 for testing.

2.3 Task Breakdown of JSA

AutoJSA subdivides the JSA process into three key tasks: Hazard Identification, Consequence identification, and Control Measure Generation. This division reflects the actual workflow followed during safety analysis, where each task builds upon the previous one to comprehensively assess and mitigate risks associated with a job.

Task 1 — Hazard Identification: This task identifies potential hazards for each job step. The inputs include the job name, job content, job step name, and step details. The output is a list of hazards associated with the specific step.

Task 2 — Consequence Identification: This task assesses the potential consequences of each identified hazard. The inputs include the same information as in Hazard Identification, plus the specific hazard. The output is a list of possible consequences.

Task 3 — Control Measure Generation: This task generates control measures to mitigate the identified hazards. The inputs include the job name, job content, job step details, hazard, and consequence. The output is a set of control measures.

Table 1 shows the number of samples in the training

set, validation set, and test set for each task.

Table 1: Distribution of samples across training, validation, and test sets for each task.

Task	Sample Size		
	Train	Validation	Test
Task 1	241	38	30
Task 2	759	117	92
Task 3	759	117	92

2.4 Application of LLM

LLMs are advanced AI systems trained on extensive textual data to generate human-like text and perform tasks like text generation, translation, and question answering. In this study, LLMs automate and improve the JSA process, aiding in hazard identification, consequence identification, and control measure generation. Fine-tuning and RAG techniques are used to optimize the model for better accuracy and relevance in the JSA domain.

Considering the task's complexity and resource requirements, we use GLM-4-9B-Chat [19] as the base model. GLM-4-9B-Chat is a large language model with approximately 9 billion parameters, developed as part of the GLM-4 series by Zhipu AI. By fine-tuning and applying RAG, we adapt the model for JSA report generation, improving its ability to identify hazards, assess consequences, and suggest control measures.

2.4.1 Fine-tuning

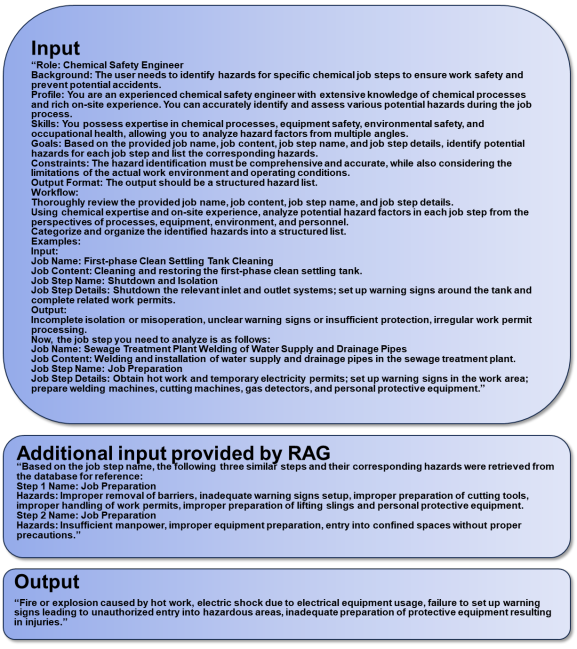

Figure 3: A sample from the dataset used in Task 1. The data used in this study is in Chinese. For the convenience of reading, all related content here and in subsequent sections has been translated into English.

Fine-tuning involves adapting a pre-trained LLM to a specific task or domain by continuing its training on a smaller, domain-specific dataset [20]. While pre-trained models like LLMs are trained on large, diverse datasets to capture general linguistic patterns, fine-tuning helps the model specialize in a particular domain—in this case, JSA. This approach improves the model's ability to identify hazards, assess consequences, and generate control measures, ensuring that the outputs are relevant and accurate for the specific task.

Fine-tuning works by initializing the pre-trained model with knowledge from its original training, followed by further training on task-specific data. In addition to task-specific inputs (as described in Section 2.3), prompt engineering [21] is also employed to guide the model during training. **Figure 3** illustrates an example dataset used for Task 1, and similar datasets were created for Tasks 2 and 3. The corresponding datasets can be found in the supplementary materials. Fine-tuning is performed by adjusting parameters based on the loss values from the training and validation sets, and the model with the lowest validation loss is selected for testing.

2.4.2 RAG

RAG enhances LLMs by incorporating external retrieval mechanisms to provide domain-specific, up-to-date information [22]. Unlike traditional models that rely solely on pre-trained knowledge, RAG retrieves relevant external data to improve accuracy. In the RAG framework, the model retrieves documents based on the input query, enhancing generation for more contextually appropriate responses, which is crucial for tasks like JSA.

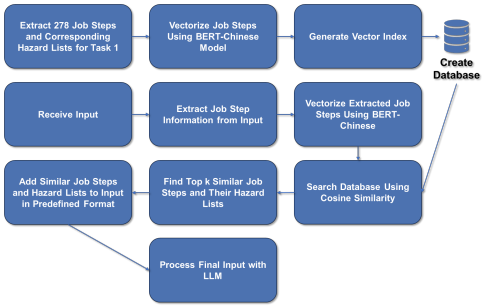

Figure 4. RAG framework uesd in this study.

For this study, we augmented the model's inputs with content retrieved via knowledge retrieval, as shown in **Figures 3** and **4**. For Task 1, a knowledge database was created by extracting 278 job steps and hazard lists. Using the BERT-base-Chinese model [23], we vectorized the job steps and indexed them. The model retrieves the **k** most similar job steps based on cosine similarity and incorporates the corresponding hazard lists into the input for the LLM. Similar processes were applied to Tasks 2 and 3, with variations in the search fields and content retrieved, and the RAG database used in this study is

available in the supplementary materials.

We evaluated different values of **k** to determine the optimal performance for each task, selecting the value that performed best on the validation set.

2.4.3 Evaluation Criteria

Model performance was assessed by measuring semantic similarity between model outputs and ground truth. Our similarity evaluation employs a BERT-based semantic encoding scheme: texts are first encoded using the BERT-base-Chinese model (768-dimensional hidden states), where the **[CLS]** token's hidden state serves as sentence embeddings. After L2-normalization, we compute cosine similarity through $\mathbf{sim}(a,b)=a^{\top}b$ to quantify semantic similarity.

3. RESULTS AND DISCUSSION

3.1 Fine-tuning

We adjusted the fine-tuning parameters based on the model's loss performance on the validation set. After multiple rounds of experimentation, we identified a set of reasonable parameters that resulted in a lower validation loss. The specific parameters used can be found in the supplementary materials. **Figure 5** illustrates the training and validation loss curves for the three tasks under these parameters during model training.

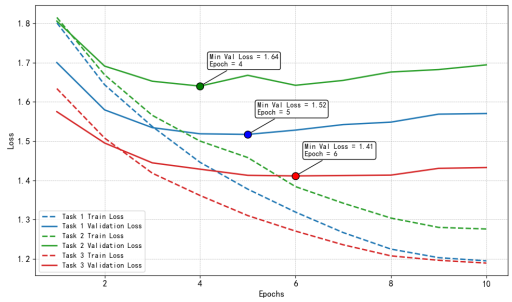

Figure 5. Training and validation loss for three tasks.

From the data trends, under the parameter settings used in this study, the training and validation losses for all three tasks showed a decreasing trend as the number of training epochs increased. This indicates that fine-tuning was effective for all three tasks. However, after a certain number of epochs, the validation loss for each task started to increase, suggesting overfitting. Based on these experimental results, we selected the models corresponding to the epochs with the lowest validation loss — 5, 4, and 6 epochs for Tasks 1, 2, and 3, respectively — to be tested on the test set.

3.2 RAG

In the construction of the RAG framework, we optimized the value of **k** for each task based on the validation set scores. **Figure 6** shows the model's scores on the validation set for the three tasks with different values of **k**, along with the time consumption per output generation.

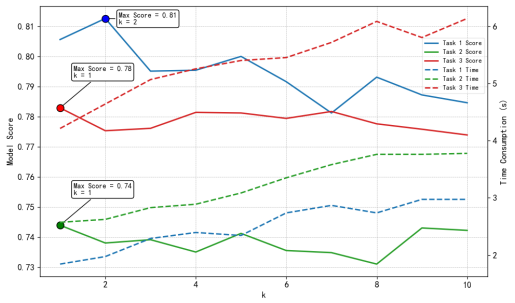

Figure 6. Model scores and time consumption vs. **k**.

During the RAG framework construction, the model's performance did not exhibit a significant trend with varying **k** values. However, when looking at all three tasks, smaller values of **k** already yielded good results. This suggests that the LLM places more importance on the most similar inputs, and it is more effective at leveraging the most relevant knowledge to solve the problem. As the value of **k** increases, more knowledge did not result in improved model performance. This could be due to the inclusion of more low-similarity knowledge, which may have interfered with the model's reasoning. Additionally, as **k** increases, the input text becomes longer, leading to longer inference times. Based on these observations, we selected **k**=2, **k**=1, and **k**=1 for Tasks 1, 2, and 3, respectively, for testing on the test set.

3.3 Test Results

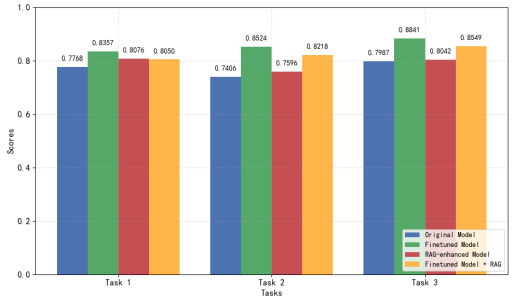

Figure 7. Different model scores on the test set.

Figure 7 shows the performance scores of the original model, the fine-tuned model, and the RAG-enhanced model on the test set for the three tasks. It also includes the results of the fine-tuned model when further augmented with RAG.

The test results show that both the fine-tuned and RAG-enhanced models outperform the original model. However, RAG provides modest improvements, while fine-tuning offers more substantial gains. Fine-tuning requires more computational resources and a complex

Model	Output	Score
Original model	Incomplete isolation or incorrect operation, unclear warning signs or insufficient protection, irregular handling of work tickets, improper equipment preparation, risk of gas leakage, fire or explosion caused by electric welding sparks, burns from high temperature operations, injuries from improper operation of cutting machines, improper use of protective equipment, poor on-site ventilation, risk of chemical exposure, noise pollution, and dust pollution.	**0.7135**
Fine-tuned model	Hot work risk, temporary electricity risk, inadequate equipment preparation, insufficient protective equipment.	**0.8477**
RAG-enhanced model	Improper handling of hot work and temporary electrical work permits, incorrect placement of warning signs, inadequate equipment preparation, insufficient preparation of gas detectors and protective equipment, insufficient personnel training.	**0.7725**
Expected	Fire or explosion caused by hot work, electric shock due to electrical equipment usage, failure to set up warning signs leading to unauthorized entry into hazardous areas, inadequate preparation of protective equipment resulting in injuries.	**-**

Table 2: Comparison of output results.

implementation, while RAG is simpler and allows dynamic updates to the knowledge base, offering flexibility.

When we combined the fine-tuned model with RAG, no performance improvement was observed over fine-tuning alone. This suggests that a more advanced integration approach, such as using a separate model to combine the outputs of both, may be needed to fully leverage the strengths of both techniques for optimizing JSA report generation.

3.4 Case Study

Table 2 shows the results of the original, fine-tuned, and RAG-enhanced models for hazard identification, as described in **Figure 3**. The original model provides a broad list of potential hazards in wastewater plant piping welding, covering a wide range of risks such as gas leaks, fire hazards, and equipment malfunctions. While comprehensive, this list can be overwhelming and less relevant to the task, requiring users to filter key risks.

The fine-tuned model provides a more focused output, listing four primary hazards: hot work operations, temporary electrical risks, inadequate equipment preparation, and insufficient protective equipment. This approach improves clarity and helps users quickly identify critical hazards, although it may overlook secondary risks. The RAG-enhanced model incorporates external databases, identifying five specific hazards, including non-compliance with safety permits and improper equipment placement. This model offers a detailed and practical identification of risks, but its effectiveness depends on the quality of the external data used.

In summary, the original model offers broad coverage but may lead to information overload. The fine-tuned model provides precise and efficient results, ideal for quick hazard identification. The RAG-enhanced model ensures comprehensive hazard identification with external knowledge. A better approach may involve strategically combining fine-tuning and RAG to enhance accuracy and detail.

4. CONCLUSION

The results demonstrate that the AutoJSA framework enhances the accuracy and efficiency of Job Safety Analysis by leveraging fine-tuning and RAG with LLMs. Both methods outperform the base model in hazard identification, consequence assessment, and control measure generation, with fine-tuning generally providing more significant improvements. The fine-tuned model produces more concise outputs, while the RAG-enhanced model generates more comprehensive results.

However, the study has limitations. The current framework does not explicitly incorporate traditional risk assessment factors such as hazard severity and probability of occurrence. The dataset of 73 Chinese JSA reports, while sufficient for proof of concept, may not fully represent the diversity of tasks in broader industrial settings or different linguistic contexts. Fine-tuning, though effective, involves high computational cost, and combining it with RAG did not yield additional benefits, suggesting potential inefficiencies in their integration. Additionally, the BERT-based similarity metric used may not capture all qualitative aspects of JSA effectiveness.

Future research directions could address current limitations by integrating risk assessment metrics into the framework, expanding the dataset to encompass diverse industrial scenarios across different linguistic and cultural contexts, exploring integration strategies for RAG and fine-tuning to enhance complementary benefits, and developing hybrid evaluation metrics that combine BERT-based similarity with qualitative safety performance indicators.

AutoJSA demonstrates AI's transformative potential in workplace safety management by enhancing hazard identification, consequence Identification, and control measure generation. The framework paves the way

toward fully automating JSA reports and ensuring safer operational practices in complex industries.

DIGITAL SUPPLEMENTARY MATERIAL

The datasets, fine-tuning parameters, and all experimental results in this study can be accessed via the following link:
https://cloud.tsinghua.edu.cn/d/f80c1ca00d694f7ea7f9/

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China under Grant 2024YFC3013600.

REFERENCES

1. Analysis JH, Lebowitz J. Essentials of job hazard analysis. Chem Eng Prog 114:52-6 (2018).
2. Albrechtsen E, Solberg I, Svensli E. The application and benefits of job safety analysis. Saf Sci 113:425-37 (2019).
3. Ghasemi F, Doosti-Irani A, Aghaei H. Applications, shortcomings, and new advances of job safety analysis (JSA): findings from a systematic review. Saf Health Work 14(2):153-62 (2023).
4. Swartz G. Job hazard analysis: A guide to identifying risks in the workplace. Government Institutes (2001).
5. Geronsin R. Job hazard assessment: a comprehensive approach. Prof Saf 46(12):23 (2001).
6. Yoon IK, Seo JM, Jang N, Oh SK, Shin D, Yoon ES. A practical framework for mandatory job safety analysis embedded in the permit-to-work system and application to the gas industry. J Chem Eng Jpn 44(12):976-88 (2011).
7. Palega M. Application of the job safety analysis (JSA) method to assess occupational risk at the workplace of the laser cutter operator. Manag Prod Eng Rev 12(3) (2021).
8. Roughton J, Crutchfield N. Job Hazard Analysis: A Guide for Voluntary Compliance and Beyond. Butterworth-Heinemann (2015).
9. Zheng W, Shuai J, Shan K. The energy source-based job safety analysis and application in the project. Saf Sci 93:9-15 (2017).
10. Chi NW, Lin KY, Hsieh SH. Using ontology-based text classification to assist job hazard analysis. Adv Eng Informatics 28(4):381-94 (2014).
11. Namian M. Factors Affecting Construction Hazard Recognition and Safety Risk Perception. North Carolina State University (2017).
12. Li W, Cao Q, He M, Sun Y. Industrial non-routine operation process risk assessment using job safety analysis (JSA) and a revised Petri net. Process Saf Environ Prot 117:533-8 (2018).
13. Pandithawatta S, Ahn S, Rameezdeen R, Chow CW, Gorjian N, Kim TW. Development of a knowledge graph for automatic job hazard analysis: The schema. Sensors 23(8):3893 (2023).
14. Zhang S, Boukamp F, Teizer J. Ontology-based semantic modeling of construction safety knowledge: towards automated safety planning for job hazard analysis (JHA). Autom Constr 52:29-41 (2015).
15. Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, ... McGrew B. GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023).
16. Bernardi ML, Cimitile M, Pecori R. Automatic job safety report generation using RAG-based LLMs. In: 2024 International Joint Conference on Neural Networks (IJCNN). Ed: IEEE (2024) p. 1-8.
17. Uddin SJ, Albert A, Ovid A, Alsharef A. Leveraging ChatGPT to aid construction hazard recognition and support safety education and training. Sustainability 15(9):7121 (2023).
18. Kvale DK. Deep Learning in Construction Safety: Quality Assessment, Hazard Identification, and Preventive Measure Proposals in Job Safety Analysis. NTNU (2023).
19. GLM T, Zeng A, Xu B, Wang B, Zhang C, Yin D, ... Wang Z. ChatGLM: a family of large language models from GLM-130B to GLM-4 all tools. arXiv preprint arXiv:2406.12793 (2024).
20. Dodge J, Ilharco G, Schwartz R, Farhadi A, Hajishirzi H, Smith N. Fine-tuning pretrained language models: weight initializations, data orders, and early stopping. arXiv preprint arXiv:2002.06305 (2020).
21. White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, ... Schmidt DC. A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv preprint arXiv:2302.11382 (2023).
22. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, ... Kiela D. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst 33:9459-74 (2020).
23. Devlin J. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).