A Large Language Model-based Method for Standardizing Heterogeneous Risk Registers

Stefano Marchetti[a], Somil Varshney[a], Cristian Schaad[a], Adrian Maker[a] and Katrina M. Groth[a]
[a]Systems Risk and Reliability (SyRRA) Lab, Center for Risk and Reliability, Reliability Engineering, University of Maryland, College Park, 20742, MD, USA.
Emails: smachet@umd.edu, somilv@umd.edu, cschaad@umd.edu, amaker@umd.edu and kgroth@umd.edu

Abstract: Risk registers are documents used to report identified risks, their potential causes and consequences, assigned owners, and planned mitigation actions. They are widely used in safety and project risk management to support risk tracking, accountability, and decision-making. In the nuclear sector, risk registers support the systematic identification, prioritization, monitoring, and communication of technical, operational, and regulatory risks throughout the lifecycle of facilities, systems, and projects. However, risk registers are typically stored in heterogeneous, human-readable free-text formats that are difficult to use for downstream analytics and machine learning. Traditional Natural Language Processing (NLP) approaches, such as rule-based pipelines, template matching, and supervised information extraction models, often exhibit limited robustness when applied to documents with variable structure, inconsistent terminology, and scarce labeled data, requiring extensive manual engineering or domain-specific annotation to generalize effectively. To overcome these limitations, we present a Large Language Model (LLM)-based method for converting diverse risk registers into a unified machine-readable schema, developed as the winning solution to the OECD Nuclear Energy Agency (NEA) Coding Competition. The proposed approach combines deterministic parsing, used wherever information can be reliably extracted from structured inputs, with an LLM-based pipeline for higher-level semantic tasks. The core contribution is a multi-agent architecture in which LLM agents are assigned specific roles and tasks, including schema mapping for unfamiliar spreadsheets, enrichment of missing metadata, disambiguation of repeated or underspecified descriptions, and conservative typo correction. To guide the behavior of the agents and improve results consistency, a few-shot learning approach based on representative input-output examples is employed. Overall, this hybrid approach achieves high accuracy by restricting LLM generative inference to tasks that require semantic reasoning, while preserving deterministic, rule-based consistency wherever structure can be directly recovered. The method was evaluated on heterogeneous risk registers, achieving approximately 90% accuracy in information extraction and standardization, showing generalization across formats and domains. More broadly, this work demonstrates how LLM agents can be effectively integrated with deterministic pipelines to build scalable and auditable document intelligence systems.

1. Introduction

Risk management is a central activity in safety-critical industries, where organizations must systematically identify, assess, prioritize, and control potential sources of harm, performance degradation, regulatory non-compliance, and project delay [1]. Within risk management frameworks, risk registers are used to record identified risks together with related information such as causes, consequences, responsible owners, and mitigation actions [2]. Risk registers are widely used because they provide a practical and traceable basis for communicating risk information among technical teams, managers, regulators, and other stakeholders [3].

Despite their practical value, risk registers are often difficult to use for downstream analytics. In practice, they are commonly stored as spreadsheets, tables, reports, or other semi-structured documents developed by different organizations and teams [4]. As a result, they may vary substantially in structure, terminology, level of detail, and completeness. Similar information may be distributed across different fields, expressed using project-specific language, or embedded in free-text descriptions [3]. These inconsistencies make it difficult to convert risk registers into standardized, machine-readable datasets

that can support querying, comparison, statistical analysis, machine learning, or integration with other risk assessment tools.

Traditional natural language processing and document-processing methods provide only partial solutions to this problem. Early information extraction systems often rely on rule-based methods, including regular expressions, dictionaries, gazetteers, manually defined ontologies, and template- or wrapper-based extraction rules [5]. These approaches can achieve high precision when the input format and terminology are stable, but they are unreliable when applied to documents with unfamiliar layouts, inconsistent column names, merged fields, or project-specific language [6]. Statistical and supervised information extraction methods, including named entity recognition [7], relation extraction [8], slot filling [9], and sequence-labeling models such as conditional random fields [10], recurrent neural networks [11], and transformer-based encoders [12], offer greater flexibility. However, these methods generally require annotated training data and task-specific labels, which are often scarce, expensive, or unavailable in specialized safety domains [13]. More recent layout-aware document understanding models, such as LayoutLM-type architectures [14], incorporate textual and spatial layout information and have improved extraction from scanned forms and document images. Nevertheless, they typically require suitable training or fine-tuning data and may still be sensitive to changes in document structure, optical character recognition quality, and domain terminology. As a result, conventional natural language processing approaches often require substantial manual engineering, domain-specific annotation, or model adaptation before they can generalize across heterogeneous risk registers [15].

Large Language Models (LLMs) offer new opportunities for extracting and standardizing information from diverse technical documents because they can interpret context, map unfamiliar terminology to target concepts, and reason over incomplete or inconsistent text [16]. However, directly applying LLMs to safety-related documents also introduces challenges [17]. Fully generative extraction may produce inconsistent outputs, alter source information unnecessarily, or infer content that is not explicitly supported by the input. For this reason, LLM-based document processing methods for risk applications should be designed to preserve traceability, consistency, and auditability.

To address these challenges, this paper presents a hybrid method for converting heterogeneous risk registers into a unified machine-readable schema. The method was developed as the winning solution to the OECD Nuclear Energy Agency (NEA) Coding Competition. The proposed approach combines deterministic parsing, used wherever information can be reliably extracted from the input structure, with an LLM-based pipeline for tasks that require semantic interpretation. These tasks include mapping unfamiliar spreadsheet columns to a common schema, enriching missing metadata, resolving underspecified descriptions, and applying conservative typo correction.

The main contribution of this work is a multi-agent architecture in which LLM agents are assigned specific roles within the extraction and standardization workflow. Rather than relying on a single unconstrained LLM call, the pipeline decomposes the problem into targeted subtasks and guides the agents through representative input-output examples using few-shot learning. This design restricts generative reasoning to cases where semantic interpretation is needed, while preserving deterministic consistency wherever the document structure can be directly recovered.

The method was evaluated on heterogeneous risk registers and achieved satisfactory accuracy in information extraction and standardization. These results show that hybrid deterministic-LLM pipelines can generalize across variable formats while maintaining the transparency needed for risk related applications. More broadly, this work demonstrates how LLM agents can be integrated with deterministic document processing to support scalable and auditable transformation of human-readable risk information into structured datasets for downstream analytics and decision support.

The remainder of the paper is organized as follows: Section 2 describes the proposed methodology, Section 3 presents the case study, Section 4 shows the results of the application and in Section 5, conclusions are drawn.

2. Proposed methodology

The proposed methodology converts heterogeneous risk registers into a standardized machine-readable format using a hybrid deterministic-LLM pipeline. The method takes as input: i) one or more human-readable risk registers and ii) a target output schema defining the desired structure of the standardized register. The risk registers may differ in file format, layout, terminology, scoring convention, and level

of completeness. The target schema specifies the output fields to be produced, such as risk identifier, risk description, project stage, project category, risk owner, mitigation action, likelihood, severity, and risk priority.

Information that can be recovered directly from the document structure, such as numerical scores and explicitly provided text fields, is extracted using deterministic rules. LLM-based inference is reserved for tasks that require contextual reasoning, including schema mapping, missing-field enrichment, disambiguation of underspecified descriptions, typo correction, and extraction from unstructured or weakly structured text.

This separation is important for safety-related document processing. Risk registers often contain information that should be preserved as faithfully as possible, such as mitigation actions, owners, and numerical assessments. A fully generative approach could unnecessarily rewrite these fields or infer unsupported information. Conversely, a fully rule-based approach is brittle when applied to heterogeneous documents with inconsistent layouts and terminology. The proposed method therefore combines the reproducibility of deterministic parsing with the flexibility of LLM-based semantic processing.

2.1. Problem formulation

Let the input be a set of N risk register documents $D = \{d_1, \dots, d_i, \dots, d_N\}$ and a target schema $S = \{s_1, \dots, s_j, \dots, s_M\}$, where each s_j represents a required or optional output field. The objective is to transform each input document d_i into a structured table whose rows correspond to individual risk entries and whose columns conform to the target schema S.

In this formulation, standardization involves three operations. First, the method must identify risk entries within the source document. Second, it must map source fields, which may use different names or structures, to the corresponding target fields. Third, it must normalize values so that equivalent information is represented consistently across registers. This includes harmonizing likelihood and severity scales, standardizing risk priority labels, and preserving additional information when it is relevant to the target schema or useful for downstream analysis.

2.2. Overview of the standardization pipeline

The pipeline consists of two main stages: deterministic extraction and LLM-based semantic processing. Figure 1 shows a schematic representation of the proposed standardization pipeline. In the first stage, the method extracts information that can be obtained directly from the source document. For structured or semi-structured spreadsheets, this includes reading cell values, identifying headers, mapping known columns, converting scores, and preserving additional fields. When a known schema is detected, a dedicated deterministic parser is used. When a schema is unfamiliar, the pipeline uses the LLM only to infer the mapping between source columns and target fields; the row-level extraction is then still performed deterministically.

In the second stage, the pipeline identifies fields that remain missing, ambiguous, or semantically underspecified after deterministic extraction. These cases are passed to task-specific LLM agents. Each agent performs a narrowly defined task and returns a structured output that can be parsed and applied programmatically. The LLM is therefore used as a semantic support component rather than as an unconstrained generator of the final register.

The final output is a standardized table aligned with the target schema. This workflow is designed to maintain traceability while enabling generalization across heterogeneous risk register formats.

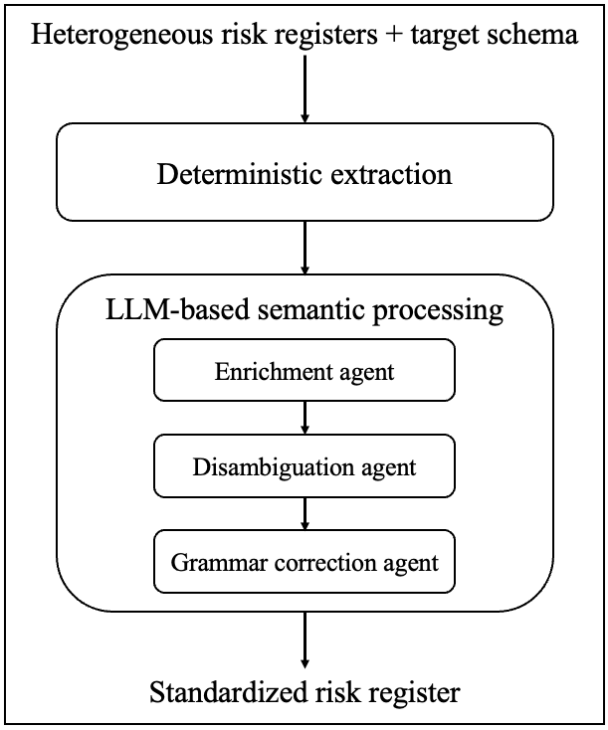

Figure 1. Steps of the proposed methodology.

2.3. Deterministic information extraction

The deterministic component is responsible for extracting and transforming information that is explicitly represented in the input. For structured spreadsheets only, this includes reading source columns and assigning them to target fields, renaming headers, preserving source text, and converting numerical values where required. This stage avoids unnecessary LLM intervention and ensures that directly available information remains traceable to the source document.

The deterministic parser can also perform document-level and record-level transformations when these are required by the target schema. For example, if a source register contains duplicate or repeated representations of the same risk, the parser can apply a predefined deduplication rule. If a register contains multiple assessment stages, such as initial and residual risk ratings, the parser can preserve these as separate fields rather than collapsing them into a single value. More generally, the deterministic stage handles any transformation that can be specified through transparent rules.

Risk scoring is also harmonized at this stage. Because source registers may use different likelihood and severity scales, the method maps all scores to the scale required by the target schema. Quantitative scores are converted using predefined scaling rules, while qualitative labels are converted using fixed lookup tables. For example, labels such as “rare,” “unlikely,” “possible,” “likely,” and “almost certain” can be mapped to increasing values on a numerical likelihood scale, while severity labels can be treated analogously.

When risk priority is not already available in a reliable form, it is computed from the product of likelihood and severity:

$$R = L \times S \tag{1}$$

where L is the standardized likelihood score and S is the standardized severity score. The resulting score is classified as Low, Medium, or High using calibrated thresholds. When an input register already provides a risk priority or risk index that matches the expected output convention, that value is used directly.

Text normalization is also performed deterministically when appropriate. Depending on the target output requirements, the pipeline may remove line breaks, normalize repeated spaces, or preserve original formatting. These operations are controlled by parser-specific settings so that text cleaning is applied only when it improves standardization without altering the meaning of the original entry.

2.4. LLM-based semantic processing

The LLM component of the method is implemented as a set of task-specific agents. Each agent receives a tailored prompt, operates on a limited subset of the data, and returns a constrained output that is parsed programmatically. In the implementation evaluated in this work, all LLM calls used Claude Sonnet 4 (claude-sonnet-4-20250514) with the temperature set to 0 and a maximum output length of 8192 tokens. All LLM responses were requested in structured JSON format. This constraint was used to support automatic parsing, reduce post-processing ambiguity, and ensure that model-generated values could be applied only to the intended fields. When appropriate, the prompts include representative input-output examples following a few-shot prompting strategy, which has been shown to improve LLM performance on task-specific outputs [18]. The full prompt templates used in the implementation are provided in Appendix A, while representative case-study-specific input-output examples used for few-shot prompting are reported in Section 3

The first LLM task is semantic enrichment. After deterministic parsing, the pipeline checks whether each row is missing critical fields, such as project stage, project category, or risk owner. The model is asked to infer concise and consistent labels from the risk description and, where available, the mitigation action, without overwriting existing values. This design ensures that the LLM supplements deterministic extraction rather than replacing it. To improve consistency, the enrichment prompt includes representative examples showing how risk descriptions should be mapped to project stages and project categories. For example, risks involving permits and licenses are mapped to regulatory or planning categories, whereas procurement delays are mapped to construction or procurement contexts. This approach promotes consistent classification across heterogeneous registers without requiring supervised model training.

The second LLM task addresses repeated or underspecified descriptions. Some registers contain multiple risks with the same generic description but different risk names. In these cases, the deterministic parser identifies the repeated descriptions and passes them to the LLM together with the corresponding risk names. The prompt includes representative examples showing how risk-name context can be used to add a concise qualifier to the original description. The model is instructed to add only the clarifying context needed to distinguish the risks, while preserving the original description text as much as possible. This step improves the usefulness of the standardized output without allowing unrestricted rewriting.

The third LLM task performs typo correction. The typo-correction agent receives batches of text fields and is instructed to fix only obvious spelling errors. Unlike the enrichment and disambiguation tasks, this step is primarily controlled through strict prompting rules rather than few-shot semantic examples. The prompt explicitly prohibits changes to meaning, grammar, punctuation, capitalization, and spacing. This design prevents the correction step from becoming a general rewriting step and helps preserve fidelity to the original register.

3. Case study

The case study consists of five heterogeneous risk registers provided in different formats and organizational structures. The risk registers come from engineering, public-sector project delivery, cybersecurity, infrastructure, and corporate risk-management contexts. The objective is to convert each input register into a standardized machine-readable output following a common target schema, which includes the core fields required to represent a risk register in a structured form: risk identifier, risk description, project stage, project category, risk owner, mitigating action, likelihood, severity, and risk priority. When available in the source register, additional information such as date, comments, action status, results, and separate pre- and post-mitigation assessments is retained in the standardized output. For this case study, risk priority is assigned using a case-study specific convention: risk scores below 20 are classified as Low, scores of 56 or greater are classified as High, and all remaining scores are classified as Medium.

The case study was designed to test whether the proposed pipeline could generalize across diverse document layouts, terminology, scoring conventions, and degrees of completeness. Some registers were already organized in a structure similar to the target schema, while others required column remapping, score harmonization, metadata enrichment, deduplication, or extraction from weakly structured text. This variability makes the dataset suitable for evaluating the intended division of labor between

deterministic parsing and LLM-based semantic processing. Table 1 summarizes the input registers used in the case study.

Table 1. Input risk registers.

ID	Risk register type	Number of risk entries	Brief description
1	Engineering project risk register	32	Risk register for an engineering deployment project, including design, procurement, commissioning, operation, and decommissioning risks
2	Public-sector project risk register	45	Local-government project risk register covering project delivery, resources, stakeholders, and mitigation actions
3	Cybersecurity risk register	3	IT and digital-security risk register covering infrastructure, access management, backup, and cyber-intrusion risks
4	Infrastructure project risk register	10	Construction/infrastructure risk register using qualitative likelihood and severity ratings
5	Corporate risk register	20	Organization-level risk register covering strategic, operational, financial, governance, IT, and continuity risks

The engineering project risk register is the most complex structured input. Its information is distributed across a non-trivial tabular layout, with separate groups of fields describing risk identification, risk assessment, mitigation, residual assessment, and related planning information. The register also includes repeated or near-repeated risk information, requiring consolidation before generating the standardized output. This input therefore tests the ability of the pipeline to combine deterministic parsing with targeted semantic processing for missing or underspecified fields.

The public-sector project risk register is closer to a conventional risk register. It already includes many of the fields required by the target schema, including risk identifiers, descriptions, project stages, categories, risk scores, owners, and mitigation-related information. The main challenge is therefore not extracting information from an unstructured source, but correctly interpreting source-specific fields and preserving values that are already present in usable form. This input tests whether the pipeline can avoid unnecessary LLM intervention when deterministic extraction is sufficient.

The cybersecurity risk register introduces domain-specific cybersecurity terminology and multiline textual fields. Its risks refer to topics such as identity and access management, backup and recovery, infrastructure, and cyber intrusion. This input is useful for assessing whether the pipeline can preserve technical content while using the LLM only to infer concise missing metadata labels.

The infrastructure project risk register introduces a different standardization challenge because its likelihood and severity assessments are expressed using qualitative labels rather than numerical scores. These labels have to be mapped to the numerical scale required by the target schema. At the same time, some metadata fields are incomplete and require LLM-based enrichment after deterministic score harmonization.

The final register is a weakly structured corporate risk register presented as a multi-page tabular document. It includes risk descriptions, mitigation measures, risk owners, actions, comments, and both initial and current risk assessments. Because the tabular structure is not reliably preserved during text extraction, this input requires a stronger role for LLM-based structured extraction before the entries can be normalized into the target schema. The document also includes a residual-risk heat map, illustrating that the original register combines tabular risk records with visual risk-summary information.

For this case study, representative input-output examples, reported in Table 2, were included in the LLM prompts to align the model outputs with the target schema and with the terminology used in the reference standardized registers. These examples were used for two tasks: semantic enrichment and description disambiguation. They were not used to replace deterministic extraction, but to guide the LLM when fields were missing or when source descriptions were underspecified.

Table 2. Examples used to guide LLM-based semantic processing in the case study.

Input risk description	Expected output
Appropriate permits and licenses procured in time	Project Stage: Pre-construction; Project Category: Regulations
Timing of generator procurement from IKM could be tight	Project Stage: Construction; Project Category: Procurement
Buoyancy System fails some or all of validation tests	Project Stage: Commissioning; Project Category: Design
Identity and Access Management (IAM) systems compromised	Project Stage: Operations; Project Category: Cybersecurity; Risk Owner: Infrastructure Manager

4. Results

The proposed methodology was applied to the five heterogeneous risk registers described in Section 3. The pipeline generated standardized machine-readable outputs for all input registers, producing 110 standardized risk entries in total. The outputs followed the target schema while preserving additional source-specific information when available, including dates, comments, action status, results, and separate pre- and post-mitigation assessments.

The standardized outputs were evaluated according to the grading procedure defined by the competition organizers. The evaluation combined two components. First, an automated comparison algorithm compared the submitted output files with the corrected reference risk registers; differences in mandatory fields reduced the automated score. This score was scaled to 100 and accounted for 60% of the final weighted score. Second, domain experts performed a blind ranking of the submitted outputs based on their similarity to the corrected reference registers. This expert-assessment component was also scaled to 100 and accounted for the remaining 40% of the final weighted score. The final score was therefore computed as a weighted combination of the automated comparison score and the expert-assessment score. Under this evaluation procedure, the proposed pipeline achieved an overall weighted score of 90/100 on the five-register case study. This result indicates that the hybrid deterministic–LLM strategy produced outputs closely aligned with the corrected standardized registers across different layouts, terminology, scoring conventions, and levels of structure, while limiting LLM inference to fields requiring semantic interpretation.

4.1. Representative standardization examples

Table 3 reports two representative input-output transformations produced by the pipeline. In the standardized output fields, L/S/priority denotes the likelihood, severity, and risk-priority classification, respectively.

Table 3. Representative input-output transformations produced by the pipeline.

ID	Input fragment	Standardized output fragment
1	**Risk name**: “Turbine Procuement”; **description**: “Fabricated components have significant lead times”; **owner**: “R. Tyler (lead engineer)”; **baseline FRQ/SEV**: 4/3; **residual FRQ/SEV**: 2/3	**Risk description**: “Turbine: fabricated components have significant lead times”; **Project stage**: Construction; **Project category**: Procurement; **Risk owner**: Lead engineer; **Pre-mitigation L/S/priority**: 8/6/Med; **Post-mitigation L/S/priority**: 4/6/Med
5	**Reference** 4; **risk**: “Failure of IT systems”; **risk if no action**: impact 5, likelihood 5, score 25; **current risk**: impact 4, likelihood 3, score 12;	**Risk ID**: 4; **Risk description**: “Risk: Failure of IT systems. Effects: Failure to secure and manage data leads to loss of corruption of inaccuracy of data, results in disruption to services and breaches of security. A further consequence could be financial penalties and reputational risk”; **Project stage**: Operations;

	owner: Carol Pilson / Peter Catchpole	**Project category**: IT; **Risk owner**: Carol Pilson / Peter Catchpole; **Pre-mitigation L/S/priority**: 10/10/High; **Post-mitigation L/S/priority**: 6/8/Med; mitigation, actions, and comments retained

These examples illustrate two complementary uses of the proposed pipeline. In the engineering project risk register, the source entry contains structured information that can be extracted deterministically, including the owner and the baseline and residual frequency and severity scores. The pipeline preserves these values, scales the scores from the source convention to the target 1-10 scale, and computes the corresponding pre- and post-mitigation priorities. At the same time, LLM-based processing is used to clarify the repeated generic description by adding context from the risk name, producing the standardized description "Turbine: fabricated components have significant lead times."

The corporate risk register example illustrates the use of the pipeline on a weakly structured source entry. The input record for "Failure of IT systems" contains the risk description, unmitigated and current risk scores, mitigation measures, owner, actions, and comments in a multi-column document layout. The pipeline extracts these elements into the target schema, converts the 1-5 likelihood and severity values to the 1-10 scale, and preserves additional fields such as mitigation, actions, and comments.

Together, the two examples show the integration between deterministic and LLM-based processing. Directly available values, such as identifiers, owners, scores, and mitigation text, are preserved through deterministic extraction and normalization. LLM-based processing is used only where semantic interpretation is required, such as enriching missing metadata, clarifying underspecified descriptions, or structuring information from weakly organized source documents.

Overall, the results indicate that the proposed hybrid approach can standardize heterogeneous risk registers with high accuracy while preserving the traceability of source information. Deterministic processing provided reliable extraction and normalization for structured fields, while LLM-based agents supplied flexibility for missing, ambiguous, or weakly structured information. This division of labor was essential to achieving robust performance across the five registers without relying on a fully generative extraction process.

5. Conclusions

This paper presented an LLM-based method for converting heterogeneous risk registers into a standardized machine-readable schema. The approach was designed to preserve information that can be extracted directly from the source documents, while using LLM-based inference only for tasks requiring semantic interpretation, such as schema mapping, missing-field enrichment, description disambiguation, typo correction, and extraction from weakly structured text.

The method was evaluated on five heterogeneous risk registers. The results show that the proposed pipeline can standardize registers with different layouts, terminology, scoring conventions, and levels of structure, achieving approximately 90% accuracy in information extraction and standardization. Deterministic parsing was most effective for explicitly available fields and score harmonization, while LLM agents were most useful for ambiguous, missing, or weakly structured information.

Overall, the study shows that LLMs can support risk-register standardization when embedded within controlled and auditable workflows rather than used as unconstrained generators. Future work will focus on evaluating the method on larger collections of safety-related documents, improving validation of LLM-generated fields, and integrating standardized risk-register data with downstream analytics and machine learning tools.

Acknowledgements

The authors acknowledge the OECD Nuclear Energy Agency (NEA) for organizing the coding competition that motivated this work and for providing the case-study materials used to evaluate the proposed methodology.

Appendix A. LLM prompt templates

This appendix reports the LLM configuration and prompt templates used in the case-study implementation. Curly-bracketed terms indicate dynamically generated content inserted at runtime.

A.1. Semantic enrichment prompt

Purpose: infer missing project stage, project category, and risk owner fields after deterministic parsing.
System prompt:
"You are a risk management expert. For each risk entry below, fill in the missing fields. Use short, consistent labels.
For Project Stage, choose from: Pre-construction, Construction, Construction / Commissioning, Commissioning, Operation, Operations, Decommissioning, or combine with ' / ' if multiple apply.
For Project Category, choose a concise label such as: Planning, Design, Construction, Procurement, Services, Regulations, Legislation, Financial, Surveys, Programme, Infrastructure, Cybersecurity, Stakeholder Management, Transport, Environment, Governance, IT, Health & Safety, Quality, Materials, or similar.
For Risk Owner, infer from context a reasonable role title.
Return a JSON array where each element has:
{"index": <1-based>, "Project Stage": "...", "Project Category": "...", "Risk Owner": "..."}
Only include fields that were listed as missing. Return ONLY the JSON array.

User prompt:
Fill in the missing fields for these risks:
{items}"

A.2. Description disambiguation prompt

Purpose: clarify repeated or underspecified descriptions using the associated risk name.
System prompt:
You are editing risk descriptions for a risk register. Some risks share the same generic description. For each, combine the Risk Name context with the description to make it unique and clear.
Rules:
- Use the format 'Component: description' when appropriate
- Keep descriptions lowercase after the colon
- Preserve the original description text exactly — only add a qualifier prefix from the Risk Name
- If the Risk Name info is already in the description, just capitalise appropriately
- Do NOT fix typos or change wording
Examples from ground truth:
- Risk Name "Structural Assembly Procurement", Desc "Fabricated components have significant lead times" → "Structural Assembly: fabricated components have significant lead times"
- Risk Name "Turbine Procuement", Desc "Fabricated components have significant lead times" → "Turbine: fabricated components have significant lead times"
- Risk Name "SCADA Procurement", Desc "components are not available, integration..." → "SCADA components are not available, integration..."
- Risk Name "power and Data cable procurement", Desc "Fabrication delays and out of spec delivered product" → "Power Cables: fabrication delays and out of spec delivered product"
- Risk Name "Dirveline Procurement", Desc "Custom components require fabrication lead time" → "Driveline procurement: custom components require fabrication lead time"
- Risk Name "Buoyancy System Procurement", Desc "components are not available" → "Components are not available"
Return a JSON array: [{"index": N, "description": "..."}]

User prompt:
{items_as_json}

A.3. Typo-correction prompt
Purpose: conservatively correct only obvious spelling errors in selected text fields.
System prompt:
You are a proofreading assistant. Fix ONLY the most obvious spelling errors in the numbered texts below.
STRICT RULES:
- ONLY fix misspellings that are clearly unintentional typos in important words (e.g. 'Genertor'→'Generator', 'scheudled'→'scheduled')
- Do NOT change spacing of ANY kind (keep spaces before commas, double spaces, trailing spaces, etc. EXACTLY as-is)
- Do NOT change punctuation of ANY kind
- Do NOT change capitalization
- Do NOT change grammar, word forms, or verb tenses (e.g. keep 'Consults' as 'Consults', not 'Consult')
- When in doubt, leave the text UNCHANGED
- Be VERY conservative — only fix errors you are 100% sure about
Return a JSON object mapping the 1-based index (as a string) to the corrected text. Only include entries that had errors. If no errors, return an empty object {}.
Return ONLY the JSON object.

User prompt:
Fix typos in these texts:
{numbered}

A.4. Weakly structured document extraction prompt
Purpose: extract structured risk entries from weakly structured text obtained from a multi-page risk register.
System prompt:
You are a data extraction assistant. You will receive raw text extracted from a corporate risk register PDF. Extract ALL risk entries into a JSON array. Each entry must have these fields:
- reference: the risk reference number (integer)
- risk_and_effects: the full 'Risk' and 'Effects' text combined
- impact_pre: pre-mitigation impact score (integer 1-5)
- likelihood_pre: pre-mitigation likelihood score (integer 1-5)
- score_pre: pre-mitigation score (integer, = impact × likelihood)
- mitigation: the mitigation measures text
- impact_post: post-mitigation impact score (integer 1-5)
- likelihood_post: post-mitigation likelihood score (integer 1-5)
- score_post: post-mitigation score (integer)
- risk_owner: name(s) of the risk owner
- actions: actions being taken to manage the risk
- comments: comments and progress of actions
Return ONLY a JSON array. No other text.

User prompt:
Extract all risk entries from this corporate risk register PDF text. Be thorough, do not skip any risks.
{full_text}

References

[1] International Organization for Standardization, "ISO 31000:2018 Risk management — Guidelines," International Organization for Standardization, Geneva, Switzerland, International Standard ISO 31000:2018, 2018. [Online]. Available: https://www.iso.org/standard/65694.html
[2] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 6th ed. Newtown Square, PA, USA: Project Management Institute, 2017.

[3] M. C. Leva, N. Balfe, B. McAleer, and M. Rocke, “Risk registers: Structuring data collection to develop risk intelligence,” *Saf. Sci.*, vol. 100, pp. 143–156, 2017, doi: 10.1016/j.ssci.2017.05.009.
[4] J. P. O’Har, C. W. Senesi, and K. R. Molenaar, “Development of a Risk Register Spreadsheet Tool for Enterprise- and Program-Level Risk Management,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2604, no. 1, pp. 19–27, 2017, doi: 10.3141/2604-03.
[5] S. Sarawagi, “Information Extraction,” *Found. Trends Databases*, vol. 1, no. 3, pp. 261–377, 2008, doi: 10.1561/1900000003.
[6] P. Kluegl, M. Toepfer, P.-D. Beck, G. Fette, and F. Puppe, “UIMA Ruta: Rapid Development of Rule-Based Information Extraction Applications,” *Nat. Lang. Eng.*, vol. FirstView, pp. 1–40, Oct. 2014, doi: 10.1017/S1351324914000114.
[7] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investig.*, vol. 30, no. 1, pp. 3–26, 2007, doi: 10.1075/li.30.1.03nad.
[8] K. Detroja, C. K. Bhensdadia, and B. S. Bhatt, “A survey on Relation Extraction,” *Intell. Syst. Appl.*, vol. 19, p. 200244, 2023, doi: 10.1016/j.iswa.2023.200244.
[9] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, “A Survey of Joint Intent Detection and Slot Filling Models in Natural Language Understanding,” *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–38, 2022, doi: 10.1145/3547138.
[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, in ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. [Online]. Available: https://dl.acm.org/doi/10.5555/645530.655813
[11] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, 2016, pp. 260–270. doi: 10.18653/v1/N16-1030.
[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.
[13] Z. Tan *et al.*, “Large Language Models for Data Annotation and Synthesis: A Survey,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 930–957. [Online]. Available: https://aclanthology.org/2024.emnlp-main.54/
[14] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, “LayoutLM: Pre-training of Text and Layout for Document Image Understanding,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1192–1200. doi: 10.1145/3394486.3403172.
[15] X. Ma and E. Hovy, “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, 2016, pp. 1064–1074. doi: 10.18653/v1/P16-1101.
[16] D. Xu *et al.*, “Large language models for generative information extraction: a survey,” *Front. Comput. Sci.*, vol. 18, no. 6, pp. 1–24, 2024, doi: 10.1007/s11704-024-40555-y.
[17] Z. Ji *et al.*, “Survey of Hallucination in Natural Language Generation,” *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023, doi: 10.1145/3571730.
[18] T. B. Brown *et al.*, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf