Bulk Processing of Engineering Management Documents Based on Domain Question-Answering Models

Shanhong He*
Suzhou Nuclear Power Research Institute Co.,Itd
Shenzhen, Guangdong, China
128051@qq.com

Abstract

This study aims to address the challenge of bulk processing of unstructured documents in engineering project management. A document delivery batch quality inspection method that integrates rule-based natural language processing and large language model technology is proposed. A domain document question-answering model based on RAG+LLM is constructed, and document category templates are automatically formed through LDA and semantic similarity clustering methods. This supports management organizations in establishing standards and enables intelligent quality judgment and key information extraction, effectively enhancing document management efficiency and accuracy, reducing manual quality inspection workload, and providing an effective technical solution for the bulk automated processing of unstructured documents in the field of construction project management. It holds significant practical value for improving project management standards.

CCS Concepts

• **Applied computing** → Document management and text processing; Document management.

Keywords

Engineering Project Management, Unstructured Document Processing, Large Language Models, RAG, LLM

ACM Reference Format:
Shanhong He. 2025. Bulk Processing of Engineering Management Documents Based on Domain Question-Answering Models. In *2025 International Conference on Artificial Intelligence and Computational Intelligence (AICI 2025), February 14–16, 2025, Kuala Lumpur, Malaysia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3730436.3730488

1 Introduction

Digital transformation has spurred significant progress in engineering project management. Tools such as Building Information Modeling (BIM), Document Management Systems (DMS), and project management software have transformed how materials are created, stored, and shared among stakeholders [1] [2]. These technologies enhance material management efficiency and information exchange accuracy. Throughout project stages, numerous unstructured documents are generated. In construction engineering's design and initiation, there are design documents, proposals, etc. During process management, logs, reports, etc., are produced. In acceptance and delivery, as-built drawings and other documents emerge [3].

Unstructured document extraction technology is evolving from closed, rule-based to open, rule-less extraction, and from simple supervised to deep-learning-based large models. This aims to increase processing diversity, improve extraction accuracy and efficiency, and integrate information into decision-support systems [4]. In natural language processing, various methods extract knowledge from unstructured texts. Xia et al. [5] noted supervised learning's data-hungry nature in entity-relationship extraction. Mintz et al. [6] proposed distant supervision, though it has noisy data. Surdeanu et al.'s [7] MIML and Ratinov et al.'s [8] self-training methods boost model robustness. Wang Zijia et al. [9] used BERT for knowledge extraction, Zeng et al. [10] for relationship classification. To address ambiguity, external knowledge sources like knowledge graphs are used. Liu et al. [11] used BERT's context-awareness, Peng et al. [12] a domain-adaptive BERT, and Wang et al. [13] domain-specific resources. Yet, these need much domain-labeled data and face term evolution challenges.

Construction project management deals with large-scale, diverse-format unstructured documents. Existing extraction technologies require type-specific training, causing high difficulty and cost. This paper proposes a batch quality inspection method for document delivery, integrating rule-based NLP and large language model technologies for intelligent quality judgment and key information extraction without type-specific training.

2 Algorithm Design

Given the diverse types of unstructured documents in project management, constructing a document feature extraction model with professional domain knowledge is essential. Pretrained language models have addressed issues like poor generalization, limited knowledge coverage, long construction cycles, and high-dimensional complexity in traditional natural language processing models. In text processing, they have proven beneficial for various downstream NLP tasks. However, general models lack enterprise private and domain-specific knowledge, thus unable to reason according to personalized business needs during enterprise implementation. Therefore, fusion-enhanced retrieval and generation methods should be adopted to optimize models' domain cognitive ability and form domain document question-answering models.

Based on this, we can quickly classify documents by their content characteristics to determine the engineering stage and document

*Corresponding author

AICI 2025, Kuala Lumpur, Malaysia

ACM ISBN 979-8-4007-1363-7/2025/02
https://doi.org/10.1145/3730436.3730488

type. For scenarios with clear management norms, use the required document categories in the norms as a classification basis. Set questions and input them into the domain document question-answering model to extract information and perform classification matching. In scenarios without management norms, adopt an unsupervised clustering method based on content characteristics. After identifying distinguishable document types, define categories based on their similar characteristics. Then, set document information extraction question-answering strategies according to the automatically defined categories. After document classification, use the domain document question-answering model to perform targeted key extraction tasks based on each category's key information items, and evaluate the effectiveness of extraction considering the overall context.

2.1 Domain Document Question-Answering Model

The RAG (Retrieval-Augmented Generation) library plays a crucial role in building large language models for domain question answering. It combines large language models (LLM) with data sources in specific domains, enabling the models to access the latest business data, thereby improving the accuracy and relevance of their answers to questions within specific domains. RAG technology supports LLM solutions through real-time data access, protects data privacy, and alleviates the hallucination problem of LLM.

2.1.1 Document Segmentation. Document segmentation is a vital step in constructing the RAG (Retrieval-Augmented Generation) library. Long texts need to be broken into smaller segments to fit into the embedding model, given the maximum token length limits of many large language models. This ensures the segmented text chunks fit within the LLM (Large Language Model)'s context window while preserving semantic integrity, improving processing efficiency.

Typical segmentation methods include character-based (CharacterTextSplitter), token-based (TokenTextSplitter), and those based on clustering algorithms. When selecting a segmentation algorithm, consider the model's token limit, semantic integrity, processing efficiency, and application scenarios. The model's token limit sets the maximum length of segmented chunks. Semantic integrity demands that chunks retain the original text's semantics. Processing efficiency relates to the algorithm's time complexity and resource consumption, and application scenarios determine quality requirements.

For engineering project management documents, which require handling context relevance, semantic integrity is crucial. Thus, when choosing a segmentation algorithm, prioritize methods that maintain semantic integrity. The token-based segmentation method (TokenTextSplitter) is a good option as it can better preserve semantic integrity, avoiding breaks at key words or mid-sentence, which is essential for subsequent text processing. Additionally, experiment with different chunk sizes to find the optimal value, as chunk size impacts the RAG system's performance.

2.1.2 Embedding Model. During RAG library construction, the embedding model is crucial. It converts text into high-dimensional vector representations for semantic-similarity-based information retrieval. The model helps the RAG system understand the original text's semantic structure, retrieve relevant information from large datasets, and support the subsequent generation model. When choosing an embedding model, consider factors like semantic understanding, model scale, domain adaptability, training data relevance, and performance. For large-data applications, models like Gte-Qwen1.5-7B-instruct are preferred. For easy deployment and low resource consumption, Jina-embeddings-v2-base-zh is a good choice. Each model has unique advantages and applications. For engineering project management documents, prioritize a model's semantic understanding and domain adaptability. Retrieved information should closely match user queries and support accurate generation. This may involve choosing high-performing models or fine-tuning. After considering model size, video memory, and average recall rate, Bge-large-zh-v1.5, with its moderate size, memory usage, and high recall, is selected as the embedding model.

2.1.3 LLM Model. For this task, mainstream large language base models can meet requirements. Considering performance, configuration needs, and tool maturity, ChatGLM-6B is selected as the LLM base for RAG in the experiment. Its multilingual ability, moderate parameter count, efficient pre-training and alignment techniques, long-text processing capacity, open-source community support, and good performance make it an advantageous RAG base model.

Combining RAG with ChatGLM-6B fully utilizes ChatGLM-6B's deep language understanding and generation capabilities. Meanwhile, RAG's retrieval enhancement improves the question-answering system's accuracy and response quality in specific domains. This architecture boosts the model's practicality and application potential, especially in scenarios with numerous domain documents and a need for accurate question-answering support.

Common large language model deployment, operation, and management frameworks include LangChain, Text Generate Webui, Xinference, etc. These offer tools and strategies for model optimization, deployment, and auto-scaling, enabling LLMs to run efficiently on various hardware and handle large-scale real-time inference requests.

LangChain, a multifunctional tool, provides a standardized and modular approach to connect models with proprietary and external data, as well as services. It allows developers to easily interact with models and integrate various components. Given its ease of use and flexibility, this project chooses LangChain as the framework. It simplifies LLM application development and enhances application security and compliance through templates and integration with NVIDIA NeMo Guardrails.

2.2 Document Classification and Information Extraction

Based on the domain document question-answering model, document types are judged by setting questions. When there are different-category document templates, key information and content structures in templates are transformed into classification questions. These questions, carefully constructed, guide the model to focus on document key information, triggering it to retrieve and generate relevant type-related information. The questions are then input into the large language model. Through deep-learning algorithms, the model understands the questions, retrieves relevant data

from specified documents, and uses its learned knowledge to reason and extract question-relevant information. The extracted key information is matched with preset standard templates. By comparing similarity, the document type is inferred. This process, via reasonable question-setting and the model's deep-learning ability, improves document management efficiency and accuracy, enabling effective key information extraction and type identification.

In scenarios without existing templates, a series of standard questions are set to cluster document-extracted results. Then, feature recognition and annotation are done on the clustering results to define and form category templates. Multiple models and algorithms can be used for document clustering by structural and content similarities. The LDA model preprocesses documents, imports policy vocabulary, generates basic data, and uses weighted algorithms for text calculation to improve policy text clustering accuracy. Semantic-similarity-based text clustering calculates word semantic similarity for a matrix, then performs spectral clustering. The vector-space-model-based document clustering converts documents into a vector space and uses algorithms like k-means for clustering.

For this project, comparing different methods yields better performance. When mining deep document semantic information, the LDA-topic-model-based policy text clustering is preferred as it handles large-scale text data and identifies topic distribution, improving clustering accuracy. For documents with rich semantic information, semantic-similarity-based text clustering can be considered. The extended Synonym Forest can calculate word semantic similarity, mining text subject potential and enhancing clustering quality. After comprehensive comparison, LDA-topic-model-and semantic-similarity-based methods are used simultaneously for clustering, with screening based on clustering distinguishability in different scenarios.

3 Experiment Setting

3.1 Environment Configuration

In this project's experiment, a high-configuration workstation was used to test RAG domain document question-answering system for large language models. The workstation features an Intel(R) Xeon(R) Gold 6430 processor with a 2.2 GHz base frequency, 2.8 GHz turbo frequency, 12 cores, and 24 threads for high processing efficiency. To speed up AI computations, an NVIDIA GeForce RTX A6000 graphics card with 48GB GDDR6X video memory was chosen. The experiment ran under a single-node configuration, primarily using the RTX A6000 GPU for large-scale model training and inference. All models and datasets in the experiments were stored locally and accessed via high-speed SSDs to boost data reading speed. To guarantee experimental result accuracy, the system was restarted before each experiment to clear cache and temporary files, ensuring a clean environment.

3.2 Training of Domain Document Question-Answering Models

To experiment with RAG domain document question-answering system for large language models, a small database was established, encompassing the management norms and delivery standards of three typical engineering management projects. It includes multiple document formats like docx, doc, and pdf.

During the project, key project management document contents were carefully selected, covering dimensions such as file names, project names, responsible units, management processes, policies, and operational procedures. The selected content is diverse and of varying complexity, aiming to comprehensively test the system's handling of different information types and complexity levels. After choosing key sections, each section was divided into paragraphs, each with a relatively independent theme. This boosts the retrieval module's precision and ensures the generation module gets specific, context-relevant information. Maintaining paragraph logical relationships and contextual coherence is vital for subsequent modules.

Since large models struggle with internal corporate knowledge, directly processing initialization vectors can result in low retrieval hit rates. Thus, the RAG process was comprehensively optimized for better retrieval accuracy. In the experiment's preparation stage, data quality was enhanced by optimizing data indexing, including improving the index structure, refining content, adding metadata, optimizing alignment, and adopting a hybrid retrieval strategy. During retrieval, the quality of context fragments was improved by enhancing the embedding model's performance. Post-retrieval, efforts focused on addressing context window limitations, reducing noise, and improving attention.

In this experiment, the parent document backtracking method was adopted to solve issues of insufficient or redundant context information due to fixed-length traditional document chunks. This method decomposes large documents into small chunks for storage, balancing accuracy and context. The process involves: using two text splitters to create parent and child chunks; vector-embedding child chunks for accurate semantics; storing each parent chunk's complete text in memory with a unique ID; creating a document backtracking object with the vector store, memory store, and splitters. When adding a document, the parent splitter generates a unique ID stored in the memory store, and the child splitter further divides parent chunks into child chunks stored in the vector store with parent ID as metadata. During retrieval, relevant child chunks are fetched from the vector store, their parent IDs are extracted, and corresponding parent chunks are retrieved from the memory store. This design combines small and large chunk advantages, using small chunks for similarity matching and returning large-chunk results with more context, balancing semantic precision and context integrity.

3.3 Document Classification

For document classification feature extraction, a series of relevant questions were designed. For instance, referring to document types in the "Handbook for Completion Acceptance of Construction Projects" of a certain unit, which include feasibility studies, task statements, design data, management documents, foreign-related documents, production and financial data, various facility-related acceptance documents, and project completion acceptance documents. Document classification questions are: What's the document name? Which project does it belong to? What's the construction unit? What sections does it have?

Table 1: Original document data size

Project	Size	Documents (.doc)	Documents (.pdf)	Spreadsheets (.xls)
Project A	5.8GB	17	4588	136
Project B	1.2GB	0	1421	12
Project C	10.9GB	8048	9043	4437

Table 2: Document classification result

Project	Total number of documents	High-confidence match	Low-confidence match	No match
Project A	4741	4520	15	206
Project B	1433	1305	72	56
Project C	21528	20316	1078	134

Based on document-related question answers and the similarity of document names and chapter structures, a Latent Dirichlet Allocation (LDA) model-based text matching method is employed. It mines hidden topic-word relationships in the text and uses the distribution to calculate text similarity. This method is apt for handling complex-structured and semantically rich documents, effectively extracting topic features from chapter structures for similarity comparisons.

Classification experiments were conducted on the delivery materials of three typical projects. The original document data volume is presented in Table 1.

Classification tests were conducted on the delivery documents of the three projects using the domain document question-answering model, and the results are shown in Table 2.

In the results, classification results with a similarity over 0.9 to any template and not exceeding 0.6 to all other templates are defined as high-confidence matches. Those with a similarity of more than 0.6 to multiple templates are low-confidence matches, and the rest are no matches. Low-confidence and no-match documents are manually reviewed. Low-confidence matches mainly from template word ambiguity, while no-match projects often differ significantly from delivery standards or management systems, indicating document quality issues. Experimental results show that the LDA topic-model-based method can notably enhance text similarity calculation accuracy and text clustering effectiveness. It can effectively determine document-template similarity based on document chapter structures and contents, thus identifying document categories. This method exhibits high accuracy and reliability when handling complex-structured and semantically rich documents.

3.4 Document Information Extraction

For classified documents, information extraction questions are set according to the focuses of different types of templates to automate document information extraction. When designing questions, we ensure they cover various levels of details, from basic factual to complex operational ones, to comprehensively represent paragraph content. After generating the questions, we label the corresponding context paragraphs and standard answers for each. Finally, a <question, retrieval context, standard answer>-form dataset is created, as presented in Table 3.

The annotation used the double-annotation method. Two independent experts annotated each question and answer, followed by a consistency check to ensure annotation accuracy and reliability. The context paragraphs were input for the retrieval module, and the standard answers were used for comparison and analysis by the subsequent evaluation module.

Based on this data, document information extraction quality was evaluated. A group of experts conducted subjective evaluations with indicators including answer accuracy, information refinement degree, and text consistency. For each indicator, five rating options were set; a higher score indicated better performance methods for the corresponding indicator. Details of the scoring metrics shown in Table 4.

After sampling the information extraction results of the highly classified matched documents of the three projects, an evaluation of the effect of automatic information extraction by the model was carried out according to the sampling ratio of 2% of the document quantity of the three projects. The results are shown in the Table 5.

Experimental results indicate that the interactive approach with the domain document question-answering model can effectively achieve batch information extraction of engineering project management delivery materials. The key information obtained from documents via question-answering has a high correlation with the original file content, and feature extraction is stable. This extracted key document information can effectively support subsequent full-content retrieval. Moreover, by matching category templates, it can identify potential document quality issues like missing chapters, content errors, and project mismatches. This significantly reduces the manual quality inspection workload during document delivery and effectively enhances the management capabilities of document receiving units.

4 Conclusion and Future Work

This paper presents a batch quality inspection method for document delivery, integrating rule-based natural language processing and large language model technologies to address unstructured

Table 3: Sample format of the dataset.

Type	Description
Question	The question input by the user
Retrieval context	The context retrieved from external knowledge sources according to the user's question, that is, the documents related to the question
Standard answer	The real (correct) answer based on the question provided by humans

Table 4: Document Information Extraction Scoring Sheet.

Type	Description	Score
Accuracy of the answer	Whether the generated answer strictly follows the established standards, it should reflect clear causal, logical and chronological order relationships, and the summary and analysis process should be included in the answer.	1.Extremely inaccurate. The answer completely fails to meet the requirements, lacks logic, causality and chronological order, and there is no summary and analysis process. 2.Relatively inaccurate. Most of the answers do not meet the requirements, and the logical and other relationships are not clear, with obvious deficiencies in summary and analysis. 3.Generally accurate. Part of the answer meets the requirements, and the logical and other relationships are rather vague, with insufficient summary and analysis. 4.Relatively accurate. The answer basically meets the requirements, and the logical and other relationships are relatively clear, with a certain summary and analysis. 5.Extremely accurate. The answer fully meets the requirements, with clear logic, causality and chronological order, and includes detailed summary and analysis.
Degree of information refinement	When facing descriptive requirements, whether it is possible to accurately extract supplementary information from similar or noisy documents and comprehensively cover all aspects involved in the question based on comprehensively refining the document information.	1.No refinement. The model fails to extract relevant information, and the text cannot effectively answer the question. 2.Low refinement. Little information is extracted, and the coverage is limited. 3.Moderate refinement. Some information is extracted, and the answer is one-sided. 4.High refinement. The model extracts most of the information, with some details missing. 5.High-degree refinement. The model can accurately extract all relevant information, and the generated text comprehensively answers the question.
Degree of text alignment	The correlation between the generated text and the content of the provided document. It is required that the generated content is highly relevant to and closely combined with the document.	1.No alignment. The text has almost no connection with the document. 2.Low alignment. Most of the information is not closely combined. 3.General alignment. Some information is not closely combined. 4.High alignment. Most of the information is closely combined. 5.High-degree alignment. The generated text is closely combined with the original document information, and the degree of relevance is high.

Table 5: Document Information Extraction Scoring Results.

Subjective evaluation	Project A	Project B	Project C	Project D
Accuracy of the answer	4.4	4.7	4.3	4.4
Level of information refinement	4.1	4.3	4.8	4.1
Degree in text alignment	3.8	4.1	4.6	3.8

document processing in construction project management. By constructing a domain document question-answering model, it achieves intelligent document quality judgment and key information extraction. Experimental results show that this method significantly boosts document management efficiency, reduces manual

quality inspection workload, and enhances the management capabilities of document receiving units. Additionally, the study demonstrates automated formation of document category templates when none exists, improving document classification accuracy. Overall, it offers an effective technical solution for automated document processing in construction project management, with important practical value for enhancing project management transparency and accountability.

Future research will concentrate on optimizing the domain document question-answering model's performance. This includes improving information extraction accuracy and the ability to handle complex sentence structures with long-distance dependencies. Moreover, the model's application in various fields and scenarios will be explored to verify its generalization ability. The research will also focus on reducing the domain model's reliance on large amounts of labeled data and more effectively integrating external knowledge sources like knowledge graphs and domain dictionaries to aid in disambiguation and improve domain-specific term recognition. Through these endeavors, the model's practicality and application potential in real-world construction project management are expected to be further enhanced.

References

[1] F. X. Zhou, Y. Huang, and Q. L. Li, "Research on quality management of construction engineering projects from the perspective of process," Journal of Engineering Management, vol. 30, no. 01, pp. 98–102, 2016, doi: 10.13991/j.cnki.jem.2016.01.018.

[2] W. Zhong, K. L. Zhang, G. Y. Yang, *et al.*, "Research on business process reengineering of engineering project management from BIM perspective," Journal of Graphics, vol. 38, no. 06, pp. 896–903, 2017.

[3] J. K. Si, "Analysis and application of engineering project management software in construction engineering projects," Beijing University of Posts and Telecommunications, Beijing, China, 2009, Doctoral dissertation.

[4] H. Y. Zhang, "Layout analysis and table extraction of unstructured documents," Beijing Jiaotong University, Beijing, China, 2019, Doctoral dissertation.

[5] Z. Xia *et al.*, "Research review of entity relation extraction based on deep learning," in Proceedings of the 19th Chinese Conference on Computational Linguistics, Haikou, China, 2020.

[6] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in CL-IJCNLP, 2009.

[7] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in ACL, vol. 2, 2012.

[8] L. Ratinov and D. Roth, "Local learning for natural language processing," in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009.

[9] Z. J. Wang, Y. Li, and Z. K. Zhu, "Knowledge extraction of unstructured domain texts based on BERT," arXiv.org, 2023.

[10] D. Zeng, K. Liu, S. Lai, and G. Zhou, "Relation classification via convolutional deep neural network," in COLING 2014, 2014.

[11] Y. Liu *et al.*, "Fine-grained relation extraction with contextualized embeddings," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019.

[12] B. Peng *et al.*, "Domain-adaptive BERT for domain-specific sentiment analysis," in Proceedings of the 58th Annual Meeting of the ACL, 2020.

[13] M. Wang *et al.*, "Enhancing relation extraction with domain-specific lexicons," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021.