



A Method for Text Information Extraction Leveraging Large Language Models and Two-Stage Information Filtering

Lu Ping¹, Qin Bin², Wang Yiyang², Xu Yan³,
Deng Fangwei¹

(1. ZTE Corporation, Shenzhen 518057, China;
2. Shenzhen University, Shenzhen 518060, China;
3. Beijing Xingyun Digital Technology Co., Ltd., Beijing 100176, China)

DOI: 10.12142/ZTECOM.202603011

<https://kns.cnki.net/kcms/detail/34.1294.TN.20260826.1349.004.html>,
published online August 26, 2026

Manuscript received: 2024-09-20

Abstract: This paper addresses shortcomings in the current combined application of large language models (LLMs) and vector knowledge bases by proposing a novel method for text information extraction. The method integrates the powerful language understanding capabilities of LLMs with the robust matching ability of two-stage information filtering, aiming to improve both the hit rate for retrieving relevant background knowledge and the accuracy of text information extraction. Experimental results on a real-world accident investigation report dataset demonstrate that the proposed method achieves a significant 37% improvement in extraction accuracy compared to baseline methods. Additionally, comparative results across data from different industries further validate the method's strong generalization performance, suggesting its potential as an effective technical solution for other long-text information extraction tasks.

Keywords: large language models; two-stage information filtering; text information extraction

Citation (Format 1): Lu P, Qin B, Wang Y Y, et al. A method for text information extraction leveraging large language models and two-stage information filtering [J]. *ZTE Communications*, 2026, 24(3): 92 – 98. DOI: 10.12142/ZTECOM.202603011

Citation (Format 2): P. Lu, B. Qin, Y. Y. Wang, et al., “A method for text information extraction leveraging large language models and two-stage information filtering,” *ZTE Communications*, vol. 24, no. 3, pp. 92 – 98, Sept. 2026. doi: 10.12142/ZTECOM.202603011.

1 Introduction

Text information extraction is the process of identifying and extracting key information from textual data based on predefined labels. In the current era of information overload, immense value is hidden within vast amounts of data. Efficiently extracting useful information from this data has become a key research focus in fields such as computer science, data mining, and information retrieval. However, in the context of big data, traditional methods of text information extraction face significant challenges, including high redundancy and low accuracy. Thus, developing more efficient and precise extraction technologies is crucial.

In recent years, the rapid advancement of artificial intelligence technology, spurred by the surge of large language models (LLMs) such as ChatGPT, has gained global traction. Approaches based on LLMs are increasingly becoming a new trend in text information extraction research. These large models typically consist of hundreds of millions to tens of bil-

lions of parameters, offering significantly greater learning and representational capabilities compared to traditional smaller models. After being trained on extensive datasets, they can more effectively capture complex patterns and features within text data, thus enhancing the accuracy of information extraction. These methods often frame predefined labels as questions, organizing the questions and raw text data in a specific format before inputting them into the model. The extraction results are then obtained by querying the model in this structured manner.

However, due to the current limitations of LLMs in processing long texts, even models that claim to handle long texts effectively can still experience a significant drop in accuracy when dealing with ultra-long textual data in real-world vertical domain tasks, and may even exhibit phenomena such as model hallucinations. To address this issue in the task of text information extraction, researchers have proposed a solution that integrates LLMs with vector knowledge bases. The core idea of this approach is to leverage the powerful language understanding capabilities of LLMs in combination with the precise information provided by the vector knowledge base, thereby achieving complementary advantages. The approach

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. IA20231030016.

starts with vector retrieval, which extracts semantically relevant background knowledge from the knowledge base that closely aligns with the question. This background knowledge is then combined with the question and fed into the LLM. By focusing only on a limited amount of original text that is highly relevant to the question, this approach aims to improve the reliability of the model's responses.

Nevertheless, background knowledge retrieval based on semantic similarity relies on the proximity between the question and the background knowledge in semantic space. This proximity does not always ensure content relevance, leading to a low hit rate, which has become a major limitation of this method. Additionally, it is often difficult to strike a balance between the quantity and quality of candidate entries generated through a single filtering step, which ultimately impacts the overall system performance.

To overcome these limitations, this paper proposes a novel approach—text information extraction based on LLMs with two-stage information filtering. This method decomposes the question of predefined label construction into multiple retrieval terms and an inquiry term. Each retrieval term is used exclusively for the initial retrieval, and a set operation is performed on all the initial results to ensure the uniqueness of text blocks. Finally, the background knowledge obtained through the two-stage filtering process is fed into the LLM, producing high-quality outputs. The advantages of this method are highlighted in four key aspects:

- **Higher background knowledge retrieval hit rate:** By designing multiple retrieval terms for the initial retrieval, the results are more targeted and better aligned with the practical needs of the information extraction task. Since different retrieval terms focus on different aspects of the question, this method covers potentially relevant background knowledge more comprehensively, thereby improving the hit rate.
- **Increased accuracy of text information extraction:** The inquiry term is used to further filter the background knowledge matched during the initial retrieval, selecting information that is more highly correlated with the question. This process ensures that the final retrieved background knowledge is not only semantically related to the question but also more contextually aligned with the actual needs of the text information extraction task, thereby enhancing accuracy.
- **Reduced model hallucinations:** By applying two-stage filtering and re-ranking mechanisms to further screen the background knowledge, this approach ensures that the final background information passed to the LLM is highly relevant to the information extraction task, helping to reduce model hallucinations.
- **Improved system efficiency:** The two-stage filtering and re-ranking mechanisms further streamline the background knowledge that is passed to the LLM, reducing the amount of information that the model needs to process, thus lowering computational burdens and improving overall system efficiency.

2 Related Work

2.1 Evolution of LLMs

Since the introduction of deep learning algorithms in 2006, they have achieved remarkable success in image recognition and found widespread application in natural language processing (NLP). During this critical period, models such as Word2Vec^[1], Seq2Seq^[2], and the recurrent neural network (RNN) Encoder-Decoder^[3] laid the theoretical and practical foundations for the subsequent development of LLMs.

In 2017, Google's research team introduced the Transformer^[4], whose efficient parallel processing and ability to handle long-range dependencies made it feasible to build LLMs. Based on the Transformer, Devlin et al. proposed BERT^[5], which significantly improved contextual understanding through bidirectional pre-training. BERT and its derivatives, such as RoBERTa^[6] and ALBERT^[7], achieved breakthrough results in NLP tasks. BERT-BiLSTM-ATT^[8] was specifically optimized for Chinese, enhancing its performance in understanding and generating Chinese text. Pre-trained models have now become the new paradigm in NLP, ushering in the era of large-scale pre-trained models in the field of artificial intelligence^[9].

Since 2020, large-scale pre-trained models like GPT-3^[10] have demonstrated exceptional capabilities in areas such as natural language generation, conversational systems, and text summarization, thanks to their massive parameter scale and outstanding performance. The widespread application of these models has not only gained attention in academia but also shown their practicality in industries such as finance, healthcare, and law, significantly improving efficiency and advancing intelligent services. As a result, LLMs have officially entered the stage of production applications, becoming a crucial tool for enhancing productivity and marking the transition from academic research to industrial applications.

2.2 Application and Development Frameworks of LLMs

With the continuous advancement of LLMs, their applications have extensively permeated various sectors of society, offering transformative solutions. For instance, the intelligent legal system Disc-Law^[11] leverages LLMs to efficiently process legal documents and provide consultation services, significantly enhancing the efficiency and accessibility of legal services. ChatGPT, known for generating fluent and logically coherent language, has been widely used in fields such as content creation, data analysis, and content refinement. The applications of Chinese LLMs, such as Baidu Wenxin Yiyan, Zhipu Qingyan, and iFlytek SPARK, mark a new milestone in Chinese language processing capabilities.

To simplify the development and deployment of LLMs, specialized application development frameworks such as Semantic Kernel and LangChain have emerged. These frameworks provide a suite of tools and interfaces that help developers eas-

ily integrate and apply LLMs. Semantic Kernel focuses on deeply customizing semantic understanding capabilities, while LangChain streamlines the construction of complex language processing workflows through modular design. Leveraging these frameworks to integrate LLMs with vector knowledge bases lays a solid foundation for applying LLMs to domains such as emergency management.

2.3 Text Information Extraction

In the field of NLP, Named Entity Recognition (NER)^[12] is a sub-task of text information extraction, specifically aimed at identifying entities in text that have specific meanings, such as names of people and locations. Through well-established NER methods, we can extract important information from text, highlight key points, and filter out redundant background information. However, real-world text information extraction demands are often complex and diverse. Using NER for information extraction can be limited by predefined text types and entity categories. For example, in scenarios such as extracting information from emergency plans, the diversity of required information may exceed the predefined entity categories of a model. Moreover, fine-tuning models for a large amount of information can consume substantial time and resources, creating bottlenecks in practical applications. As a result, traditional NER methods often struggle to handle challenging tasks like extracting information from emergency plans.

In recent years, the rapid development of Retrieval-Augmented Generation (RAG) technology has provided a novel approach to tackling complex information extraction tasks. By incorporating relevant information retrieved from external knowledge bases into LLMs, RAG significantly enhances the accuracy of generated content while reducing the risk of producing misleading information. Building on the traditional RAG framework, several advanced methods have

been proposed to address the challenges of processing long texts. For instance, MemoRAG^[13] introduces a memory-inspired knowledge discovery mechanism, demonstrating exceptional performance in handling long texts. LightRAG^[14] integrates knowledge graphs, leveraging graph structures to represent entities and their relationships, thereby enhancing contextual awareness and mitigating the issue of fragmented answers caused by reliance on flat data structures. ChunkRAG^[15], on the other hand, segments long texts into semantically coherent chunks and utilizes LLMs to filter out irrelevant chunks prior to the generation phase. This approach minimizes the interference of irrelevant information, thereby improving the accuracy of generated responses.

Inspired by these advancements, we applied a collaborative strategy combining LLMs and vector-based knowledge bases to information extraction tasks. This approach leverages the language model's strengths in understanding while integrating precise information provided by the vector knowledge base, achieving more efficient and accurate information extraction. Additionally, drawing on the divide-and-conquer strategy underlying ChunkRAG, our method breaks down complex queries and filters content in stages, effectively addressing the limitations of LLMs in processing long texts.

3 Proposed Method

3.1 Overview

The overall architecture of the proposed method is illustrated in Fig. 1. Specifically, this approach takes the document and the information labels to be extracted as inputs. First, a vector knowledge base is constructed based on the document. Then, an LLM is employed to generate a set of retrieval terms and inquiry terms corresponding to the information labels. Subsequently, the vectorized retrieval term set is

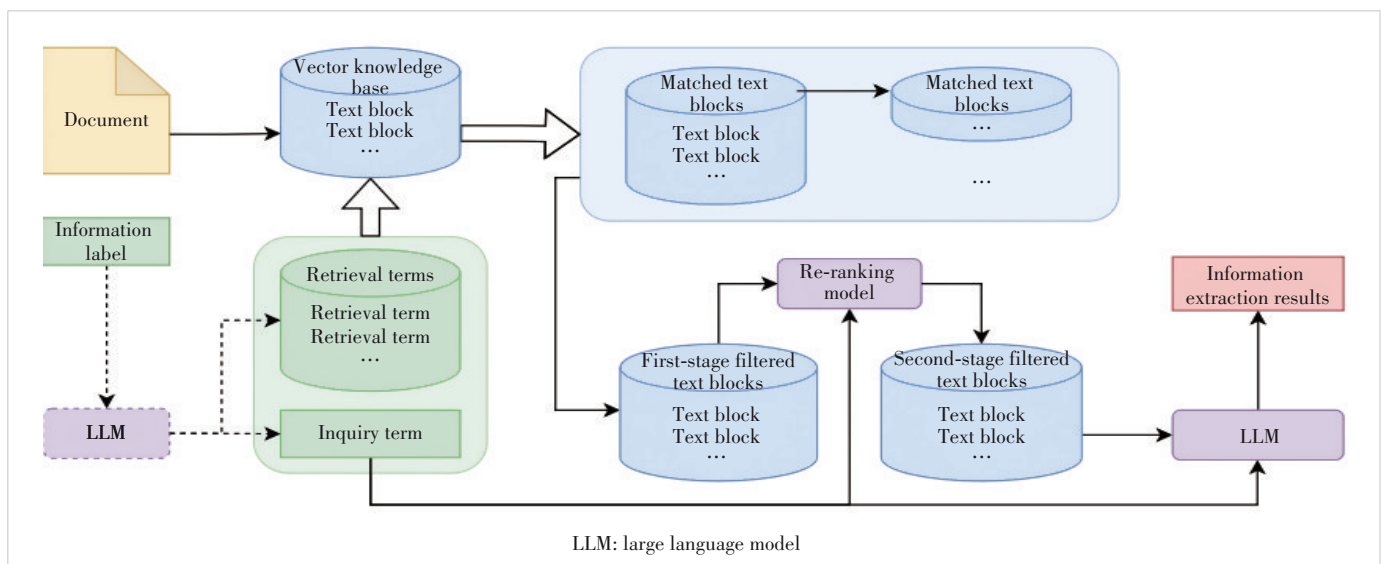


Figure 1. Framework of text information extraction method based on LLMs and two-stage information filtering

used sequentially to perform text block matching within the vector knowledge base. The resulting set of matched text blocks undergoes two-stage information filtering through semantic relevance matching with the inquiry terms. Finally, the filtered background knowledge and inquiry terms are input into the LLM, which generates the text information extraction results. The following section provides a detailed description of this process.

3.2 Vector Knowledge Base Construction

1) Long text segmentation

When processing long texts, we adopt a recursive character-based text splitting method to ensure that the length of the resulting text blocks does not exceed a predefined character limit. First, the given text is split into several text blocks at natural boundaries such as sentences or paragraphs. This initial splitting maximizes the preservation of the text's semantic integrity. Then, for each text block, if its length exceeds the character limit, the block is further recursively split until all text blocks satisfy the character limit requirement. The recursive process can be represented as the function $\text{split}(T, L)$, which accepts the text and character limit as inputs and returns a set of sub-blocks that meet the character limit:

$$\text{split}(T, L) = \begin{cases} \{T\}, & |T| \leq L \\ \bigcup_{i=1}^n \text{split}(T_i, L), & \text{otherwise} \end{cases} \quad (1)$$

where T represents the input text, L represents the character limit, and T_i represents the i -th text block obtained by splitting T .

2) Text block vectorization

At the vectorization stage, we use a pre-trained embedding model to generate high-dimensional vector representations of the text blocks. For the set of text blocks $\{T_i | T_i \in \text{split}(T, L)\}$, returned from splitting the long text, each text block T_i is converted into a vector representation.

$$\mathbf{v}_i = f(T_i), \forall T_i \in \text{split}(T, L) \quad (2)$$

where the embedding function $f(\cdot)$ is implemented by the pre-trained embedding model, which directly maps text

blocks to high-dimensional vector representations $\mathbf{v}_i$. This vectorization method preserves the semantic information of the text blocks, facilitating subsequent retrieval operations in the semantic space.

3) Knowledge base construction

When constructing the vector knowledge base, we first store the vector set $V = \{\mathbf{v}_i | T_i \in \text{split}(T, L)\}$ in the vector knowledge base M , and then build an index for each vector. This process can be expressed as:

$$M = \left(\text{store}_{\text{vectors}}(V), \left\{ \text{index}_{\text{vector}}(\mathbf{v}_i) \mid \mathbf{v}_i \in V \right\}, d(\cdot) \right) \quad (3),$$

where $\text{store}_{\text{vectors}}(V)$ denotes storing the vector set V in the database, $\{\text{index}_{\text{vector}}(\mathbf{v}_i) \mid \mathbf{v}_i \in V\}$ denotes indexing each vector, and $d(\cdot)$ represents the similarity measurement function of M . In our implementation, we use the Euclidean distance as the similarity measurement, which is defined as:

$$d(\mathbf{v}_i, \mathbf{v}_j) = \sqrt{\sum_{k=1}^m (\mathbf{v}_i[k] - \mathbf{v}_j[k])^2} \quad (4),$$

where $\mathbf{v}_i[k]$ denotes the k -th element of the text vector $\mathbf{v}_i$.

3.3 Two-Stage Information Filtering

We propose a novel background knowledge retrieval mechanism that decomposes the question into multiple retrieval terms and a separate inquiry term. By utilizing initial retrieval followed by two-stage information filtering, we enhance the retrieval hit rate of background knowledge.

3.3.1 Initial Retrieval

First, we extract multiple retrieval terms q_{search_k} from the question, with each retrieval term focusing on different aspects of the question to cover a broader range of potential background knowledge, thereby enhancing the initial search's coverage. In this method, the separation of retrieval terms and the inquiry term is aided by LLMs, with some manual intervention. For the same information label, the retrieval terms obtained from the initial separation can be embedded with the inquiry term for direct use in subsequent steps. Table 1 presents

Table 1. An example of the separation of retrieval and inquiry terms

Information Label	Before Decoupling	After Decoupling	
		Retrieval Term	Inquiry Term
Accident location	In which province, city, county (district), and street is the accident site located, and what is the exact address?	1) In which province, city, county (district), and street is the accident site located, and what is the exact address?	2) Accident location
		3) Place of the accident	4) Position of the accident
		5) Accident site	6) Area of the accident
		7) Detailed address of the accident	8) Specific location of the accident
		9) Address of the accident	...
			What is the specific location of the accident?

an example of this separation process.

Next, each retrieval term is converted into a high-dimensional vector representation $\mathbf{v}_{q_{\text{search}_i}}$:

$$\mathbf{v}_{q_{\text{search}_i}} = f(q_{\text{search}_i}) \quad (5)$$

During the similarity search process, we filter the top N results whose vector distances to each retrieval term are below the threshold S . The algorithmic formula is as follows:

$$s = \left\{ \mathbf{v}_i \mid d(\mathbf{v}_{q_{\text{search}_i}}, \mathbf{v}_i) \leq S \right\} \cap \left\{ \mathbf{v}_i \mid \mathbf{v}_i \in M \leq N \right\} \quad (6)$$

where $\mathbf{v}_i$ represents the vector representation of the text blocks stored in the vector knowledge base M .

Utilizing the similarity search algorithm s , we obtain the set of retrieval results in a vector format R_k :

$$R_k = s(\mathbf{v}_{q_{\text{search}_k}}, N, S) \quad (7)$$

We then convert the retrieved vector set R_k into the corresponding set of text blocks T_{R_k} :

$$T_{R_k} = \{T_i \mid \mathbf{v}_i \in R_k\} \quad (8)$$

Finally, we merge all text block sets T_{R_k} to ensure the uniqueness of each text block, forming a filtered set of text blocks T_R :

$$T_R = \bigcup_k T_{R_k} \quad (9)$$

Through the background knowledge retrieval mechanism that separates retrieval terms and inquiry terms, we ensure that the search results meet both the quantitative constraints and the similarity score requirements, thereby enhancing the accuracy and relevance of the retrieval.

3.3.2 Second-Stage Information Filtering

In the initial retrieval stage, we obtain a set of first-stage filtered text blocks T_R related to the retrieval term q_{search_i} . Although these results have undergone initial filtering, they may still contain a significant number of entries with low relevance. Directly passing this background knowledge in its entirety to the LLM is inappropriate, as it not only increases computational overhead but may also expose the model to noisy information, resulting in hallucinations. To further enhance the relevance of the retrieved background information, we introduce a second-stage information filtering process.

In the second-stage filtering phase, we use a pre-trained re-ranking model to further refine and rank the initially filtered background knowledge.

First, we compute the relevance score of each text block obtained from the initial filtering with respect to the inquiry term:

$$\text{score}(q_{\text{ask}}, T_i) = \text{rerank}(q_{\text{ask}}, T_i) \quad (10)$$

where q_{ask} denotes the inquiry term and T_i represents the i -th text block in the set T_R . The function $\text{rerank}(\cdot)$ is implemented using a pre-trained re-ranking model.

Next, we sort the elements of the text block set T_R based on their relevance scores and select the top K text blocks with scores above the threshold S' as the final background knowledge matching result T_{final} :

$$T_{\text{final}} = T_i \mid \text{score}(q_{\text{ask}}, T_i) \geq S' \cap T_i \mid |T_{\text{final}}| \leq K \quad (11)$$

The second-stage filtering further removes low-relevance entries from the initial filtering results, significantly improving the matching degree between the background knowledge and the inquiry term q_{ask} . This process helps us precisely target critical background knowledge, avoiding the transmission of a large volume of irrelevant information to the LLM, thus ensuring the quality and efficiency of the model's responses.

3.4 LLM Interaction

After obtaining the high-relevance two-stage filtered text block set T_{final} , we format it with the inquiry term q_{ask} using a prompt template and feed this into the LLM for final answer generation. This process can be represented as follows:

$$A = \text{LLM}(\text{template}(q_{\text{ask}}, T_{\text{final}})) \quad (12)$$

where the function $\text{template}(\cdot)$ refers to the operation of applying the prompt template, and the function $\text{LLM}(\cdot)$ is implemented by the LLM. Specifically, this method draws on the prompt engineering guidelines published by OpenAI^[16]. Based on empirical principles such as using delimiters to distinguish instructions from context and clarifying the desired output format through examples, a prompt template as illustrated in Fig. 2 has been designed. This prompt template aims to assist the LLM in better understanding the requirements of the information extraction task and generating extraction results that conform to the expected format.

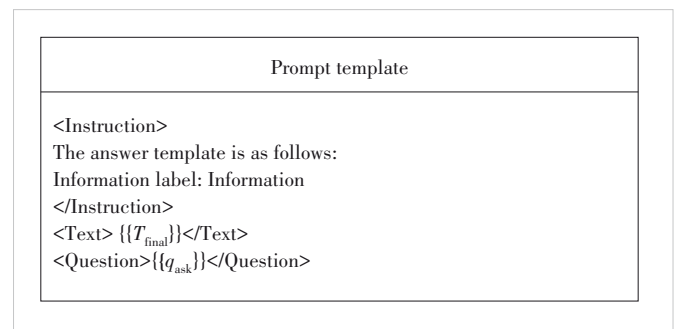


Figure 2. Prompt template for LLM

4 Experiment

4.1 Dataset

The dataset used in this study consists of 219 authentic Chinese accident investigation reports, encompassing two major industry categories: construction and commercial manufacturing, as well as several subcategories. The industry distribution is shown in Table 2.

Considering the focus on accident information in the field of emergency management, as well as the common information found in accident investigation reports, this study aims to extract four types of information from the reports, as shown in Table 3. In total, 876 information items need to be extracted from the 219 accident investigation reports.

4.2 Implementation Details

We developed an information extraction application for accident investigation reports based on the proposed method, using the LangChain and Streamlit frameworks. Fig. 3 illustrates the implementation process of the application. The selected LLM is ChatGLM3-6B^[17], which supports both Chinese and English with an 8k context length. The temperature parameter is set to

Table 2. Industry distribution of the dataset

Major Industry Category	Subdivision	Number of Reports
Construction	Residential building	48
	Civil engineering	31
Commercial manufacturing	Trade and industry	100
	Chemical industry	40

Table 3. Information labels for accident investigation reports

Index	Information Label
1	Accident location
2	Casualties
3	Economic loss
4	Legal basis

0.01. A lower temperature value means the model is more inclined to choose high-probability words during sampling, thereby generating more deterministic and consistent outputs. This aligns with the high accuracy and consistency requirements of the information extraction task. The embedding model is M3E-base^[18], the re-ranking model is BGE-reranker-v2-m3^[19], and the vector knowledge base is built using Faiss. This application is deployed on a single RTX-4090 GPU. The settings for the various parameters in the method are listed in Table 4.

4.3 Comparative Experiment

To validate the superiority of the proposed method, we conducted comparative experiments using Langchain-Chatchat^[20], a representative baseline method within the traditional LLM and vector knowledge base collaborative application framework. The blue section in Fig. 3 highlights the improvements of our method compared to the baseline. Table 5 presents the results of the comparative experiments, using accuracy as the evaluation metric.

As shown in Table 5, when extracting information from real-world Chinese accident investigation reports, the proposed method significantly outperformed Langchain-Chatchat in the

Table 4. Parameter configuration in the method

Parameter	Value
Similarity threshold S	0.5
First-stage filtering text block threshold N	20
Semantic similarity threshold S'	0.5
Second-stage filtering text block threshold K	8

Table 5. Comparative experiment results

Method	Accident Location	Casualties	Economic Loss	Legal Basis	Average
Langchain-Chatchat	0.47	0.31	0.52	0.13	0.36
Ours	0.76	0.75	0.67	0.75	0.73

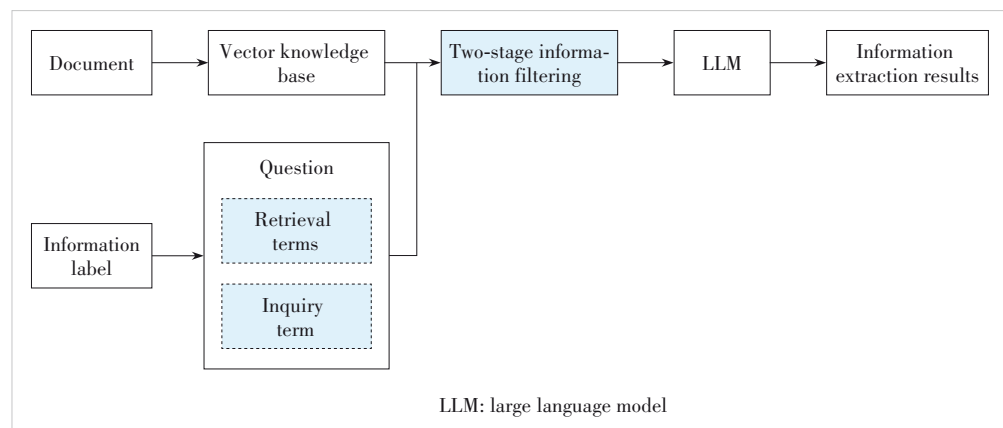


Figure 3. Flowchart of information extraction process from accident investigation reports

four information categories: location of the accident, casualty situation, economic loss, and legal basis. Notably, the accuracy for extracting legal basis improved from 13% to 75%, and the average extraction accuracy for the four categories increased from 36% to 73%.

Table 6 illustrates the differences in extraction accuracy between the two methods for accident investigation reports from various industries. This

Table 6. Experimental results across different industries

Method	Residential Building	Civil Engineering	Trade and Industry	Chemical Industry	Average
Langchain-Chatchat	0.41	0.27	0.40	0.50	0.40
Ours	0.73	0.75	0.79	0.62	0.72

indicates that the proposed method demonstrates greater generalization across different sectors.

5 Conclusions

This paper proposes a text information extraction method based on LLMs and two-stage information filtering. Validation on a real-world accident investigation report dataset demonstrates that this method significantly improves the accuracy of information extraction compared to the baseline method, while also mitigating model hallucinations. Moreover, comparative results across documents from different industries further confirm that the proposed method possesses considerable generalization capability, rather than serving merely as an optimized solution tailored to specific text types. This provides a technical reference for the practical application of long-text information extraction in other domains.

References

- [1] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [PP/OL]. arXiv [2024-09-20]. <https://arxiv.org/pdf/1301.3781v1>
- [2] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks [C]//The 28th International Conference on Neural Information Processing Systems. ACM, 2014: 3104 – 3112. DOI: 10.5555/2969033.2969173
- [3] Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [C]//The 2014 Conference on Empirical Methods in Natural Language Processing. ACL, 2014: 1724 – 1734. DOI: 10.3115/v1/d14-1179
- [4] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//The 31st International Conference on Neural Information Processing System. ACM, 2017: 6000 – 6010. DOI: 10.5555/3295222.3295349
- [5] Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [C]//Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2019: 4171 – 4186. DOI: 10.18653/v1/N19-1423
- [6] Liu Y H, Ott M, Goyal N, et al. RoBERTa: a robustly optimized BERT pre-training approach [PP/OL]. arXiv (2019-07-26) [2024-09-20]. <https://arxiv.org/abs/1907.11692>
- [7] Lan Z Z, Chen M D, Goodman S, et al. ALBERT: a lite BERT for self-supervised learning of language representations [PP/OL]. V6. arXiv (2020-02-09) [2024-09-20]. <https://arxiv.org/abs/1909.11942>
- [8] Li D Y, Tu Y F, Zhou X S, et al. End-to-end Chinese entity recognition based on BERT-BiLSTM-ATT-CRF [J]. ZTE communications, 2022, 20 (S1): 27 – 35. DOI: 10.12142/ZTECOM.2022S1005
- [9] Zhou C, Li Q, Li C, et al. A comprehensive survey on pretrained foundation models: a history from BERT to ChatGPT [PP/OL]. V4. arXiv (2023-05-01) [2024-09-20]. <https://arxiv.org/abs/2302.09419>
- [10] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners [PP/OL]. V4. arXiv (2020-07-22) [2024-09-20]. <https://arxiv.org/abs/2005.14165>
- [11] Yue S B, Chen W, Wang S Y, et al. DISC-LawLLM: fine-tuning large language models for intelligent legal services [PP/OL]. V2. arXiv (2023-09-23) [2024-09-20]. <https://arxiv.org/abs/2309.11325>
- [12] Chiu J P C, Nichols E. Named entity recognition with bidirectional LSTM-CNNs [J]. Transactions of the association for computational linguistics, 2016, 4: 357 – 370. DOI: 10.1162/tacL_a_00104
- [13] Qian H J, Liu Z, Zhang P T, et al. MemoRAG: boosting long context processing with global memory-enhanced retrieval augmentation. [PP/OL]. arXiv (2024-09-09) [2024-09-20]. <https://arxiv.org/abs/2409.05591>
- [14] Guo Z R, Xia L H, Yu Y H, et al. LightRAG: simple and fast retrieval-augmented generation [C]//Proceedings of Findings of the Association for Computational Linguistics. ACL, 2025: 10746 – 10761. DOI: 10.18653/v1/2025.findings-emnlp.568
- [15] Singh I S, Aggarwal R, Allahverdiyev I, et al. ChunkRAG: novel LLM-chunk filtering method for RAG systems [EB/OL]. V1. arXiv (2024-10-25) [2024-10-30]. <https://arxiv.org/abs/2410.19572>
- [16] OpenAI. Best practices for prompt engineering with the OpenAI API [EB/OL]. [2024-09-20]. <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api>
- [17] Du Z X, Qian Y J, Liu X, et al. GLM: general language model pretraining with autoregressive blank infilling [PP/OL]. V2. arXiv (2022-03-17) [2024-09-20]. <https://arxiv.org/abs/2103.10360>
- [18] Wang Y X, Sun Q X, He S C. M3e-base [EB/OL]. (2023-06-07) [2024-09-20]. <https://huggingface.co/moka-ai/m3e-base>
- [19] Gao L, Callan J. Condenser: a pre-training architecture for dense retrieval [PP/OL]. V2. arXiv (2021-09-20) [2024-09-20]. <https://arxiv.org/abs/2104.08253>
- [20] Chatchat-Space. Langchain-Chatchat. [EB/OL]. [2024-09-20]. <https://github.com/chatchat-space/Langchain-Chatchat>

Biographies

Lu Ping is the vice president of ZTE Corporation. He also serves as the deputy director of the State Key Laboratory of Mobile Network and Mobile Multimedia Communication. His research interests include artificial intelligence, XR, immersive communication, and network-media convergence.

Qin Bin (qinbin@szu.edu.cn) is a professor-level senior engineer of Shenzhen University, China. His research interests focus on artificial intelligence.

Wang Yiyang received her master's degree in new-generation information technology from Shenzhen University, China in 2026. Her research interests include the applications of large language models and 3D vision.

Xu Yan holds an MS degree in mechanical engineering. He is currently a product manager at Beijing Xingyun Numerical Technology Co., Ltd. His research interest focuses on the digital application of AI technology in the industry.

Deng Fangwei is a chief research engineer of ZTE Corporation. Her research interests include artificial intelligence, XR, immersive communication, and network-media convergence.